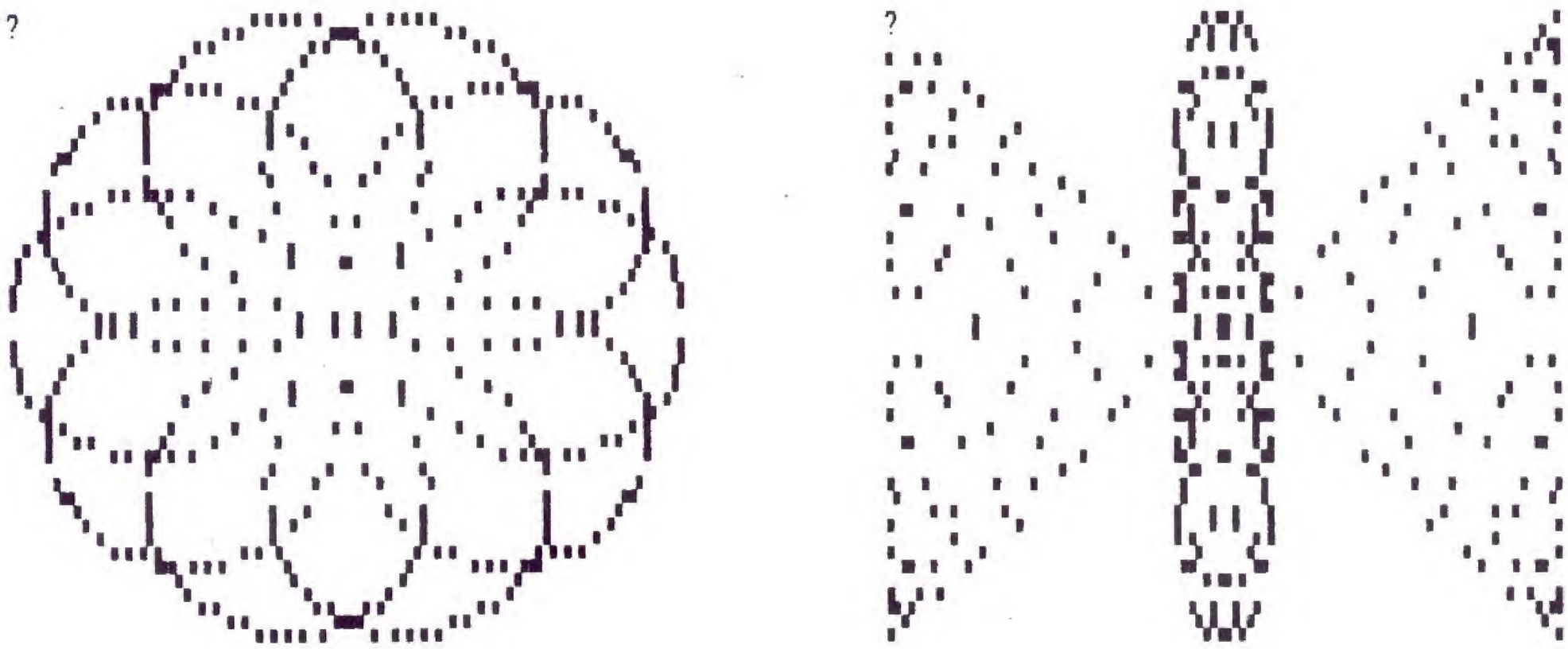
TRS-80 SYSTEM 80 VIDEO GENIE PMC-80

Vol. 3, Issue 4, March 1982



LEMNISCATES with one line of code

Also in this issue:

HARDWARE:

Simple Interface for the Model 33 Teletype.

PROGRAMMING:

The Theory and Techniques of Sorting — Part 5

It all Depends — Using Logical Operations in Conditional Statements

SOFTWARE:

- Restores to Any Line No.
- Punctuation!
- Wheel Loader Production
- Patch for Solitaire Program
- Alien Invasion

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $2.50, N.Z. $4.00, U.K. £1.50

***** ABOUT MICRO-80 *****

EDITOR:	IAN VAGG
ASSOCIATE EDITORS:	
SOFTWARE LEVEL I :	MICHAEL SVENSDOTTER
SOFTWARE LEVEL II:	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices:

	12 MONTH SUB.	SINGLE COPY
MAGAZINE ONLY	$ 26-00	$ 2-50
CASSETTE PLUS MAGAZINE	$ 65-00	$ 4-00 (cass. only)
DISK PLUS MAGAZINE	$ 125-00	$ 10-00 (disk only)

MICRO-80 is available in the United Kingdom from:
U.K. SUBSCRIPTION DEPT. 24 Woodhill Park, Pembury, Tunbridge Wells, KENT. TN2 4NW

Prices:			
	MAGAZINE ONLY	£ 16-00	£ 1-50
	CASSETTE PLUS MAGAZINE	£ 43-60	N|A
	DISK PLUS MAGAZINE	£ 75-00	N|A

MICRO-80 is available in New Zealand from:
MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 N.Z. Ph. 62894

Prices:			
	MAGAZINE ONLY	NZ$ 43-00	NZ$ 4-00
	CASSETTE PLUS MAGAZINE	NZ$ 89-00	NZ$ 5-00
	DISK PLUS MAGAZINE	NZ$ 175-00	NZ$ 15-00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB.)	MAGAZINE	CASS + MAG	DISK + MAG
PAPUA NEW GUINEA	Aus$ 40-00	Aus$ 83-00	Aus$ 143-00
HONG KONG/SINGAPORE	Aus$ 44-00	Aus$ 88-00	Aus$ 148-00
INDIA/JAPAN	Aus$ 49-00	Aus$ 95-00	Aus$ 155-00
USA/MIDDLE EAST/CANADA	Aus$ 55-00	Aus$ 102-00	Aus$ 162-00

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **
Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** LIABILITY **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR: MICRO-80 P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT: 24 WOODHILL PARK, PEMBURY TUNBRIDGE WELLS, KENT TN2 4NW

Printed by: Shovel & Bull Printers, 379 South Road, MILE END 5031

Published in Australia by: MICRO-80, 433 Morphett Street, Adelaide.

***** EDITORIAL *****

MICROSOFT COMMENCES PRODUCTION IN AUSTRALIA

MICROSOFT one of America's largest and most prestigious publishers of computer software, has commenced production in Australia in a joint venture with Wiser Laboratories. Wiser-Microsoft already has some of its CPM and Apple products in production in Sydney and is adding to its range daily. This is welcome news for TRS-80/System 80 users, since Microsoft software has been somewhat difficult to come by of late, due to long supply lines from the U.S.A. whilst the price of software on disks has sky-rocketed with the imposition of the onerous 35% duty.

A few issues ago, we stated that the Hong Kong manufacturer of the System 80/Video Genie had no plans to produce a colour computer. More recent information suggests that statement is inaccurate. We hope to be able to reveal more in the next issue.

Rumours continue to abound that the Model I TRS-80 is to be dropped from sale in Australia. This time, the strongest rumours are coming from within Tandy Australia. Like the religious sects which periodically predict the end of the earth and then sheepishly, return to their everyday business a day or so after the predition fails to materialise, we have on at least two occasions, predicted a definite end to the sale of Model I's in Australia. Probably out of a sense of spite, Tandy has proved us wrong on each occasion! This time we will wait and see.

For some time now, Tandy Australia has been selling TEAC disk drives for the Model I and Tandon drives for the Model 3. The latest drives, which we understand will be sold for both the Model I and Model 3, are a hybrid. The body (i.e. the electromechanical parts) is made by Tandon but the electronics is made by Tandy. Like MPI drives, standard Tandon drives have all the electronics necessary to run dual-heads. This is not the case with the new Tandy electronics which are unashamedly for single-sided drives only. It seems that Tandy will not be bringing out larger capacity drives for the Model 3 in the forseeable future. However, we do know that Tandy recently summonsed representatives from one of the well known DOS publishers to Fort Worth to discuss fitting 5 inch Winchester hard disk drives to the Model 3. Perhaps Tandy will go for mass storage in a big way.

- 0000000000 -

***** PEEKing (UK) by our U.K. Correspondent Tony Edwards *****

The purchase of a microcomputer is not the only hardware purchase a hobbyist or small business user makes - rather it is the first of many. Most micro-users buy a number of add-ons for their machines. Printers, disks and floppies are well known (essential) accessories but a number of others are also available. How about an '80 which talks? Various speech synthesizers are now available in this country ranging from the excellent 'Digitalker' at £120 to the equally good but more restricted 'Speakeasy' at only £69. If your '80 talks to you it is only polite for you to talk back, and a speech input interface such as 'Big Ears' will cost you about £45.

There is also a number of cheaper add-ons in the form of ROMs. The lower case and colour ROMs are well known and are available for £33 and £35 respectively, but there are a growing number of other ROMs becoming available for our '80s. GNOMIC, a U.K. company, has marketed a feature ROM at £19 which either plugs into the circuit board or fits directly into the output port of the '80 using an interface at £16 extra. This is a very interesting development. With this ROM attached the '80 gains a number of useful features, a flashing cursor, a shift lock, an auto-key repeat, single key stroke commands and a machine code editor and system tape copier. All in the hardware! I have been using this ROM for some time now and am very pleased with it. The minor features are nice to have but the machine code editor and copier have become essential. It is so convenient to have a resident machine code editor and memory copying facilities in the hardware.

What of the future? Additional feature ROMs can be expected from a number of sources and '80 owners should also look out for a crop of enhanced video add-ons. The video display on the '80 is poor and there is some room for improvement. Recently I saw a demonstration of an add-on video enhancement unit producing graphics as good as I have seen on any computer. I understand that it is suitable for use with the whole range of '80 machines and that it will be ready soon, so watch this space for further details.

- 0000000000 -

***** INPUT/OUTPUT *****

From: Mr. G. Lawrence, Heidelbereg, Victoria.

Tired of still having to load Sargon from tape even though you have disks? Well read on.

In an attempt to thwart piracy the Sargon II chess program, written by Dan and Kathe Spracklen, is recorded in a non-standard form, and is provided with its own special loader. Hence it is difficult to use a tape to disk conversion program such as Apparat's LMOFFSET as provided with their NEWDOS-80. If you own a copy of NEWDOS-80 and a copy of Eddy Paay's BMON plus at least 32K of RAM then the following procedure will overcome the problem.

1. Boot system for non-disk.
2. Set memory size to below BMON, say 44000.
3. Enter SYSTEM mode and load 32K BMON, but don't type '/'.
4. Load the special Sargon loader into memory.
5. Type /48080 to enter BMON "through the back door". In this way we do not affect reserved RAM.
6. use BMON to patch an intercept into the Sargon loader by changing...

LOC.	FROM	TO
449A	2A	C3
449B	13	D0
449C	44	BB

7. Execute Sargon loader at 4415 (hex) from BMON.
8. When Sargon is loaded you should then be returned to BMON. Now restore the modified locations back to their original contents and use BMON to punch an object tape of the first 16K of memory.

START=4000, END=7FFF, ENTRY=449A (all parameters in hex).

9. Boot the system for disk.
10. Use LMOFFSET to create a self relocating copy of the tape that you have just made, into a disk file named "SARGON/CMD". Specify a load address of 5200 (hex) and a "DOS compatible" appendage.

For the casual reader who may not be initiated into the wonders of NEWDOS-80 let me explain that LMOFFSET is a cunning little utility. Its function is to accept a memory module (or program) from tape or disk, disregard its normal load information and locate it into an area that does not clash with DOS. It then allows you to re-save the module specifying where you want it to be loaded in future. If you wish it will also add an "appendage" which will be loaded along with the module and cause it to relocate to its original location prior to execution.

To load Sargon from disk, now all you have to do is insert the appropriate diskette and type SARGON. The memory block that we saved will be loaded above DOS and then moved down to 'clobber' DOS and restore the non-disk system. Sargon will start automatically.

In closing let me say that as a software manufacturer myself, I fully sympathise with the publishers' desire to prevent piracy of their product. Unfortunately, as you can see such pirate traps are not really effective. Indeed, with tape programs the pirate doesn't even have to be very clever, all he needs to do is record the tape from one cassette player to another. My feeling is that where traps are ineffective they shouldn't be used, since they are just an inconvenience to the legitimate user. Perhaps when we all come to realise that piracy ultimately results in reduced software availability and increased prices then the practice will abate.

From: Mr. M. Bull, Traralgon, Victoria.

Over the past nine months I have been using the Microsoct Editor/Assembler-Plus package with an Exatron Stringy Floppy patch. This system has worked very well, except for the fact that although I can save SOURCE code directly to wafer, to save an OBJECT file on wafer it is necessary to save it on cassette, and then load the ESF monitor, and then reload the cassette, and then save the object file on wafer - or so I thought.

I now believe I have a way to save the object file directly to wafer, bypassing the cassette and ESF monitor completely. I think this method would be of some interest to your readers and I have listed it below.

1. Assemble the source code into memory with a manual or automatic origin.
2. Find the start, end and entry locations of the program from the assembled listing.
3. Convert these numbers from hexadecimal to decimal.
4. Calculate the length of the program.
5. Return to BASIC with either the "B" command or the reset switch.
6. Initialise the ESF (SYSTEM , /12345)
7. Use the following to save the object code:

   @(#d)SAVEn,start,length,entry

   Where #d is the drive number and is optional, and n is the file number.

   All numbers in decimal.

This will save the file with an auto start from the entry location.

- 0000000000 -

***** IT ALL DEPENDS - by Professor Brian Abrahamson *****

Few BASIC programs are written which do not contain IF....THEN....ELSE statements. Many could be simplified if logical operators were used in conjunction with these conditional statements. Unfortunately, for many of us, the combination of logical operators and conditional statements represents relatively uncharted waters and we avoid them. This article should go a long way towards helping us understand these useful programming techniques. Brian Abrahamson is the Professor of Mathematical Sciences at the Flinders University of South Australia. The article was the basis of an address to a recent meeting of the Adelaide Micro Users Group.

Once upon a time there was an apprentice sorcerer who discovered that his best friend was making time with his girl. Grim with anger, he consulted his master's books to find the WORDS OF POWER which he could use to turn his treacherous friend into a tree frog. Eventually he composed a suitable spell, but unfortunately, although he used the IF and the THEN correctly, there was a small syntax error with the ELSE - so he turned himself into a frog, and his portrait now decorates the current issue of Australian stamps.

It is no doubt case histories of this sort that fill our minds with terror when we have to write programs which involve conditional statements, like

```
IF A$ = "Y" THEN GOTO 650 ELSE GOTO 110
```

and so on. In what follows I'll try to make a few points, with illustrations, which will help make sure that the condition you think you are imposing is the same as the condition that you are actually imposing.

I believe that the root of the difficulty is a confusion of two kinds of logic: the logic of propositions, which is concerned with truth and falsity, and the logic of commands, which is concerned with what actions to take under which conditions. The first is the logic of ordinary everyday reasoning and of mathematics, while the second, officially named deontic logic, is used by computer programmers and military officers. We shall be concerned here with both propositions and commands, and some connections between them.

To begin with, what is a proposition? For the purposes of logic it is any statement which is either true or false. Thus statements like "I breathe air", " Australia is an island continent", "Adelaide is four fathoms under water", "Tokyo is in New South Wales", "2 + 2 = 4", and "1 = 2" are - all of them - propositions, while statements like "I wonder if they'll allow a casino in Adelaide", "Solve the equation 3x = 6", "All the world's his oyster" and so on, are not. We shall use small letters, like p, q, r to stand for propositions, saying that two propositions are equal, p = q, when they are either both true, or else both false.

Now you can ask your computer what it thinks about the truth or falsity of propositions merely by asking it to PRINT them. For example:

```
PRINT 1=2
```

produces the response

```
Ø
```

because 1 = 2 is a false proposition (but nevertheless a proposition). Likewise:

```
PRINT 1<2
```

results in

```
-1
```

which, believe it or not, is the computer's way of saying true. This puzzles me, because the usual convention is to let Ø stand for false and 1 for true, but it is easy enough to normalise things by multiplying the computer's output by -1; we shall then get the conventional symbolism.

There is a stock of simple propositions which the computer is designed to "understand". They are the relational statements like "2 = 3", "5 > 6", "8 <>5", "AND< AMPLE", and so on. From such as these, one can build compound propositions by using NOT, AND, and OR. So from "1 = 2" one can form NOT(1=2), and from "2 > 3" and "3 < 4" one can form (2 > 3)AND(3 < 4), or (2 > 3)OR(3 < 4). And using such compounds in their turn, we may form still more complicated compounds, for example:

```
-1 < 4 AND NOT(6 = 7) OR ((5 > 3)AND(3 > 6))
```

What do such monstrosities mean? Why, nothing, of course. Propositions don't have to mean anything; they just have to be either true or false. We have rules for telling whether such compounds are true or false, summed up in their so-called truth-tables. In these you put the relevant propositions at the heads of the columns and each row represents a different choice

of possible truth-values for the simple propositions of which the compound proposition is built up. Thus we can explain NOT quite adequately by the truth-table

p	NOT p
1	Ø
Ø	1

while the logical operators AND and OR are defined by

p	q	p AND q
1	1	1
1	Ø	Ø
Ø	1	Ø
Ø	Ø	Ø

p	q	p OR q
1	1	1
1	Ø	1
Ø	1	1
Ø	Ø	Ø

Here is a program to write the truth-table for NOT(p OR q)

```
10 '** WRITE A TRUTH-TABLE FOR TWO **
20 DEFINT P,Q
30 PRINT TAB(8)"P";TAB(12)"Q";TAB(18)"NOT (P OR Q)"
40 PRINT
50 FOR P=1 TO 0 STEP -1
60  FOR Q=1 TO 0 STEP -1
70  PRINT TAB(7)P;TAB(11)Q;TAB(17)-NOT((P=1)OR(Q=1))
80  NEXT Q
90 NEXT P
100 END
```

RUN this, and you will produce the following:

P	Q	NOT(P OR Q)
1	1	Ø
1	Ø	Ø
Ø	1	Ø
Ø	Ø	1

By varying the compound proposition, you can produce any truth-table which involves only two propositions. Incidentally, variables like p and q above are known as Boolean variables; they are capable of taking on only two values, which one can think of as true or false, yes or no, right or wrong, 0 or 1, up or down, and so on, as one finds convenient. Here is a program to write a truth-table for a compound proposition involving three Boolean variables p, q and r. The proposition used as an illustration here is p AND (NOT q OR r).

```
10 '** TRUTH-TABLE FOR THREE PROPOSITIONS **
20 DEFINT P,Q,R
30 PRINT TAB(8)"P";TAB(12)"Q";TAB(16)"R";TAB(22)"P AND (NOT Q OR R)"
40 PRINT
50 FOR P=1 TO 0 STEP -1
60  FOR Q=1 TO 0 STEP -1
70   FOR R=1 TO 0 STEP -1
80   PRINT TAB(7)P;TAB(11)Q;TAB(15)R;TAB(21)-((P=1)AND(NOT(Q=1)OR(R=1)))
90 NEXT R: NEXT Q : NEXT P
100 END
```

If you RUN it as it is, your printer will print out the following:

P	Q	R	P AND (NOT Q OR R)
1	1	1	1
1	1	Ø	Ø
1	Ø	1	1
1	Ø	Ø	1
Ø	1	1	Ø
Ø	1	Ø	Ø
Ø	Ø	1	Ø
Ø	Ø	Ø	Ø

Clearly one can build up propositions of great complexity using the simple propositions as building blocks, and the logical operators NOT, AND, OR as cement (and also a lot of brackets). When the computer evaluates such propositions, to see whether they come to -1 (and so are true) or to Ø (and so are false), it treats NOT as analogous to negation in ordinary arithmetic, so NOTp is like -x. AND is analogous to multiplication and OR to addition, so it first performs all

NOT's, then all AND's and only then the OR's. But one way to be sure of how it will respond to any given compound proposition is to put it in a truth-table program and ask the computer to tell you.

So much for truth. Now, how about action? Just as we have truth-tables for compound propositions in ordinary logic, so we have action-tables for conditional statements in deontic logic. For instance, if p is a Boolean variable (i.e. either true or false), and A denotes an action, here is the action-table for IF p THEN A:

p	RESULT
1	A
Ø	Execute the next numbered line in the program.

For the closed conditional: IF p THEN A ELSE B, where B is a second action, it reads quite simply:

p	RESULT
1	A
Ø	B

After which the computer will proceed to the next instruction (not necessarily on the next line).

The following program produces the action table for two chained IF...THEN statements, i.e. for IF p THEN IF q THEN A. Notice that the result is exactly the same as that for IF p AND q THEN A.

```
10 '** CHAINING IF...THEN'S **
20 DEFINT P,Q
30 ' ** ACTION-TABLE FOR "IF P THEN IF Q THEN DO A" **
40 PRINT TAB(8)"P";TAB(12)"Q";TAB(18)"IF P THEN IF Q DO A"
50  FOR P=1 TO 0 STEP -1
60    FOR Q=1 TO 0 STEP -1
70   PRINT TAB(7)P;TAB(11)Q;TAB(18)
80   IF P=1 THEN IF Q=1 PRINT "A": GOTO 110
90   PRINT "B"
110  NEXT Q
120 NEXT P
130 END
```

P	Q	IF P THEN IF Q DO A
1	1	A
1	Ø	B
Ø	1	B
Ø	Ø	B

The next two programs find the result of chaining together two IF...THEN...ELSE statements in two distinct ways. They each produce the action-table which is shown under them

```
10 ' **  ACTION-TABLE FOR CHAINING "IF...THEN...ELSE" COMMANDS **
20 DEFINT P,Q
30 PRINT TAB(8)"P";TAB(12)"Q";TAB(18)"RESULT"
40  FOR P=1 TO 0 STEP -1
50    FOR Q=1 TO 0 STEP -1
60    PRINT TAB(7)P;TAB(11)Q;TAB(18)
70    IF P=1 THEN PRINT "A" ELSE IF Q=1 THEN PRINT "B" ELSE PRINT "C"
80    NEXT Q
90 NEXT P
100 END
```

P	Q	RESULT
1	1	A
1	Ø	A
Ø	1	B
Ø	Ø	C

```
10 ' ** ACTION-TABLE FOR CHAINING "IF...THEN...ELSE" COMMANDS **
20 DEFINT P,Q
30 PRINT TAB(8)"P";TAB(12)"Q";TAB(18)"RESULT"
40  FOR P=1 TO 0 STEP -1
50   FOR Q=1 TO 0 STEP -1
60   PRINT TAB(7)P;TAB(11)Q;TAB(18)
70   IF P=1 THEN IF Q=1 PRINT "A" ELSE PRINT "B" ELSE PRINT "C"
80   NEXT Q
90  NEXT P
100 END
```

P	Q	RESULT
1	1	A
1	Ø	B
Ø	1	C
Ø	Ø	C

The next program prints out the action-table for three chained IF...THEN...ELSE statements, showing how the four possible actions A, B, C, D are obtained.

```
10 ' ** TRIPLE-CHAINING "IF...THEN...ELSE" COMMANDS **
20 DEFINT P,Q,R
30 PRINT TAB(8)"P";TAB(12)"Q";TAB(16)"R";TAB(22)"RESULT"
40  FOR P=1 TO 0 STEP -1
50   FOR Q=1 TO 0 STEP -1
60    FOR R=1 TO 0 STEP -1
70    PRINT TAB(7)P;TAB(11)Q;TAB(15)R;TAB(22)
80    IF P=1 THEN IF Q=1 THEN IF R=1 THEN PRINT "A" ELSE PRINT "B" ELSE PRINT "C
" ELSE PRINT "D"
90    NEXT R : NEXT Q : NEXT P
100 END
```

P	Q	R	RESULT
1	1	1	A
1	1	Ø	B
1	Ø	1	C
1	Ø	Ø	C
Ø	1	1	D
Ø	1	Ø	D
Ø	Ø	1	D
Ø	Ø	Ø	D

As an example, here is a program to convert numerals like 1, 2, 3 etc. to their ordinal forms: 1'st, 2'nd, 3'rd etc. It accepts a non-negative integer N, calculates its units digit U and its tens digit T, finds the correct termination and prints the result. N = -1 is used to signal the end of the list, and the conditional statement in line 4Ø sends us to the end of the program. If that condition is not met (i.e. if N ≠ -1) then we simply pass to line 5Ø. Lines 6Ø, 7Ø and 8Ø respectively give the conditions for the terminations 'st, 'nd and 'rd and then pass directly to the print statement in line 1ØØ (thereby by-passing the conditionals below them). Line 9Ø deals with the cases that have not been filtered out by lines 4Ø, 6Ø, 7Ø or 8Ø; these get the termination 'th.

```
10 ' ** CONVERT NUMERALS TO ORDINAL FORM **
20 DEFINT N,U,T : DEFSTR A,B
30 INPUT "INTEGER BETWEEN 0 AND 32767 INCLUSIVE ";N
40 IF N=-1 THEN 120
50 U=N-10*INT(N/10) : T=INT(N/10)-10*INT(INT(N/10)/10) : A=STR$(N)
60 IF U=1 AND T<>1 THEN B=A+"'ST" : GOTO 100
70 IF U=2 AND T<>1 THEN B=A+"'ND" : GOTO 100
80 IF U=3 AND T<>1 THEN B=A+"'RD" : GOTO 100
90 B=A+"'TH"
100 PRINT "ORDINAL FORM OF ";N;" IS ";B
110 GOTO 30
120 END
```

The next example is a program which accepts the first three letters of the names of the months, and also the number of the year (in its full four-digit form) and prints out the number of days in the month concerned. It is an interesting fact that the months can be characterised by the second and third letters of their names; February is the only one with third letter B, while all those whose second letters follow D, and whose third letters follow M have 30 days; all the other months have 31 days. The condition that the year Y be a leap year is that Y must be divisible by 4 but not divisible by 100 or else divisble by 400. This is expressed in line 5Ø, where we note that "but not" is the same as "and not" so far as deontic logic is concerned.

```
10 ' ** NUMBER OF DAYS IN THE MONTH **
20 DEFINT D,L,Y : DEFSTR M
30 INPUT "FIRST 3 LETTERS OF MONTH, 4 DIGITS OF YEAR";M,Y
40 IF M="EOL" OR Y<0 THEN 100
50 IF INT(Y/4)*4 = Y AND INT(Y/100)*100<>Y OR INT(Y/400)*400=Y THEN L=1 ELSE L=0
60 M2 = MID$(M,2,1) : M3 = RIGHT$(M,1)
65 IF M3="B" THEN D=28+L : GOTO 80
70 IF M2>"D" AND M3>"M" THEN D=30 ELSE D=31
80 PRINT "THE MONTH ";M;" OF YEAR ";Y;" HAS ";D;" DAYS"
90 GOTO 30
100 END
```

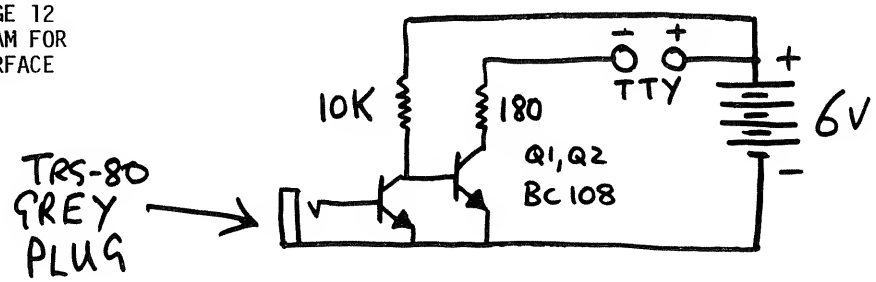
My last example is a program which paints an 8-by-8 chess-board pattern. The pixels available on the screen form an array of 48 rows and 128 columns; one has to divide them up into 64 blocks of 6 rows and 16 columns each. The key to the solution is that the pattern repeats itself every 12 rows and 32 columns. Thus if R denotes the row number and C the number of the column, the decision whether to set the pixel (C,R) or not must depend on the remainder R1 of R on division by 12, and the remainder C1 of C on division by 32. A chess-board pattern will result if one sets (C,R) when either $Ø \leqq R1 \leqq 5$ and $Ø \leqq C1 \leqq 15$ or else $6 \leqq R1 \leqq 11$ and $16 \leqq C1 \leqq 31$. This accounts for line 5Ø of the following program:

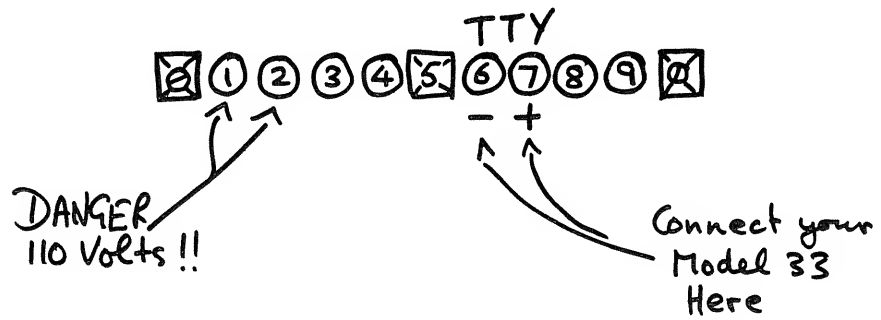
```
10 ' ** PRINT A CHESS-BOARD **
20 CLS
30 DEFINT R,C
40  FOR R=0 TO 47 : FOR C = 0 TO 127
50  R1=R-12*INT(R/12) : C1=C-32*INT(C/32)
60 IF ((0<=R1)AND(R1<6)AND(0<=C1)AND(C1<16))OR((5<R1)AND(R1<12)AND(15<C1)AND(C1<
32)) THEN SET(C,R)
70 NEXT C : NEXT R
80 FOR R=0 TO 700 : NEXT R
90 GOTO 20
100 END
```

- 0000000000 -

CONT. FROM PAGE 12

CIRCUIT DIAGRAM FOR TELETYPE INTERFACE





***** SIMPLE INTERFACE FOR THE MODEL 33 TELETYPE - by D. Harvey *****

This simple interface will allow TRS-80 users to LLIST and LPRINT programs to a Model 33 teletype machine and also print hard copy from Electric Pencil. It should only cost a couple of dollars to build and will allow you to run your Teletype from the Grey AUX cassette plug of your TRS-80. (System-80 users will have to use POKE commands to utilise the external recorder port). If you are a hobbyist with any kind of an electronic junk box you can probably make it from the bits and pieces you already have on hand. It can be assembled in about an hour by anyone with even a rudimentary knowledge of soldering techniques. The Teleprinter interface uses just the 20ma current loop part of the RS.232 connection, keeping it simpler and cheaper than other serial interfaces. Owners of Level 2, 16K machines will be able to do word processing with the Electric Pencil and other software programs without the expense of an expansion interface, though the printouts will be in upper case only.

The interface will do a simple "bare bones" job of LPRINTing and LLISTing your programs with the software included in this article, but there is a program on the market which can take your printouts to new heights of sophistication, specifying maximum and minimum line length, length for early "intelligent" termination and a host of other "goodies". It is called FORMATTER by Small System Software and costs $15. It was written for their $60 TRS-232 interface, which is similar to this interface. The address for Small System Software is P.O. BOX 366, Newbury Park, California, CA 91320, U.S.A. but you might be able to find the program locally if you're lucky.

Building the simple interface is easy. You need two small resistors, two BC-108 NPN transistors (most any type of NPN transistor will do), a 3.5mm socket to plug the grey AUX jack into and, (this is the most expensive part), an ordinary 6-volt lantern battery. The battery is your 20ma power supply, which may seem a mite crude, but since you will only have to replace it every six months or so, it's a lot cheaper than messing around with heavy (and expensive) transformers. You don't need a high voltage for as long as the teletype sees the right current it will chug along happily. Get yourself a few feet of twin flex wire in two colours so you can keep track of the polarity, a couple of small crocodile clips to hook the interface to the battery and a tiny, scrap piece of Perfboard to assemble the circuit on. A box or container for the interface is purely optional - I have used mine without any housing for more than a year with no problems other than having a wire snap off at the 3.5mm socket on one occasion.

Rummage through your junk box or (gasp!), buy:

2 NPN transistors, BC-108 (Dick Smith DS-548, or Tandy 276-2009) etc.
1 Resistor, 10 K-ohms, 1/4 W
1 Resistor, 180 ohms, 1/4 W
1 small piece of Perfboard, about 1 x 2 inches
1 miniature jack socket, 3.5mm
1 red crocodile clip
1 black crocodile clip
1 6-volt lantern battery

The following basic program will load a machine language printer-driver program into high memory, and then destroy itself. You must reserve memory space by answering the MEM SIZE? question with 32650.

```
10 REM * AUTOMATIC INTERFACE FOR MODEL-33 TELEPRINTER *
20 REM * SETS 110 BAUD, WITH LINE-FEED AND TWO NULLS AFTER *
30 REM * EACH CARRIAGE-RETURN.  OUTPUTS THROUGH GREY "AUX" PLUG *
40 POKE 16422,139:POKE 16423,127:POKE 16553,255
50 FOR I=0 TO 105:READ D:POKE 32651+I,D:NEXT I
60 POKE 32675,103:POKE 32676,2:POKE 32717,103:POKE 32718,2
70 POKE 32727,103:POKE 32728,2:POKE 32693,3:NEW
80 DATA 243,121,254,13,40,3,254,32,216,254,229,197,6,9,55
90 DATA 245,245,33,1,252,205,33,2,33,222,0,43,124,181,32
100 DATA 251,241,31,245,48,19,33,0,252,24,19,14,2,175,13,40
110 DATA 2,24,219,62,10,24,215,24,47,198,0,33,1,252,205,33
120 DATA 2,0,0,3,22,0,43,124,181,32,251,16,212,17,222,0
130 DATA 203,74,40,11,33,0,252,205,33,2,27,122,179,32,251
140 DATA 241,241,254,13,40,198,183,40,197,193,225,241,201
150 END
```

Type the program into your '80 and CSAVE it. Then use the CLOAD? facility to make sure the program was saved properly.

Now it's time to build the interface. If you aren't an electronics type, you should be able to find a friend who's a ham radio operator or other amateur electronics dabbler who'll be willing to solder the bits together for you. When you've got the board assembled, double-check all the connections and then hook up the twin-lead wire to your teleprinter. Use a twin-coloured length of twinlead to make sure you can't get the wires crossed. Hook the positive wire to screw terminal 7, and the negative wire to screw terminal 6. Terminal number 5 is an unused space without a screw, so don't let it fool you. On the left, screws 1 and 2 are carrying 110 volts mains, so beware! Don't mess around inside the teletype with the power switched on.

Once you are ready to test the teleprinter, you can plug in the mains and turn the switch clockwise to the LOCAL position. It should hum quietly and respond when you type on its own keyboard (not the TRS-80's keyboard) because it uses its own built-in 20ma power supply in this configuration. If your Model 33 works okay like a typewriter in that mode, flip the switch anti-clockwise to the LINE position. It should immediately begin chattering away like a demented cousin of R2D2, going much too fast and printing all kinds of garbage on the paper...Don't worry, that's all as it should be because the Model 33 is trying to use the external current loop which you haven't switched on yet. Now, connect the two crocodile clips to your 6 volt lantern battery. The spring in the centre is negative and the one in the top corner is the positive, which is confusing, so make quite sure which is which....If all is well, your epileptic teleprinter should calm down again, just like it did in LOCAL mode. If it still chatters away printing gibberish, turn it off, disconnect the battery from the interface, and start checking every connection. If the teleprinter remains quiet when turned on in the LINE mode, you know it is getting the 20ma current loop it needs from your el-cheapo interface and the lantern battery. You can now load up the software and test the interface.

Turn off the power to the TRS-80 and wait a few seconds before powering up again. Answer the MEM SIZE? question with 32650 and hit ENTER. Then CLOAD the program. LIST it first on your screen to check it hasn't garbled and then RUN it. After a couple of seconds the VDU screen should clear itself and then say READY. With the printer turned on and the interface connected to the grey AUX plug and the lantern battery, you are ready to try a command from your TRS-80's keyboard. Type:

LPRINT"TESTING"ENTER

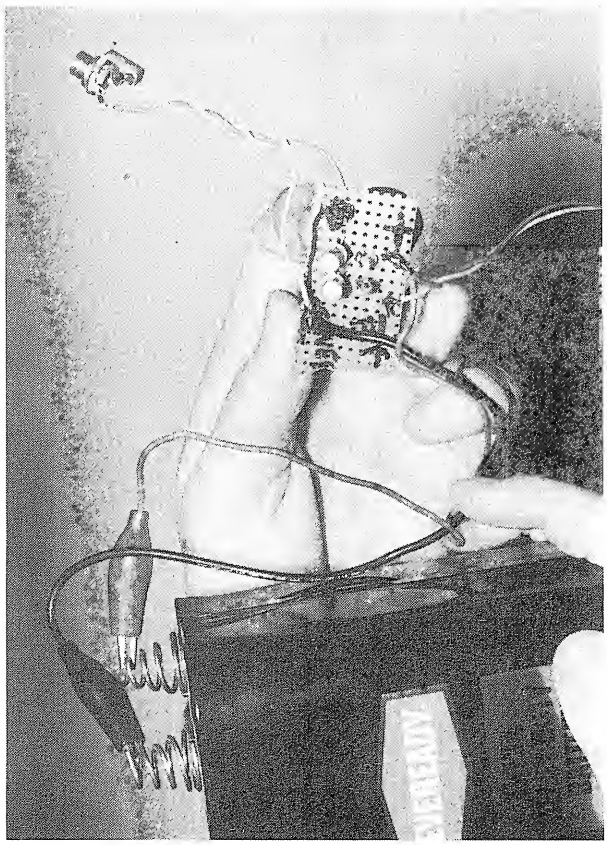
If the el-cheapo is working, the teletype will begin clattering away and the word TESTING will be typed out in hard copy. You're in business!!

If you don't have a word processing program, you might like to give this super-simple program a try - just to test the printer:

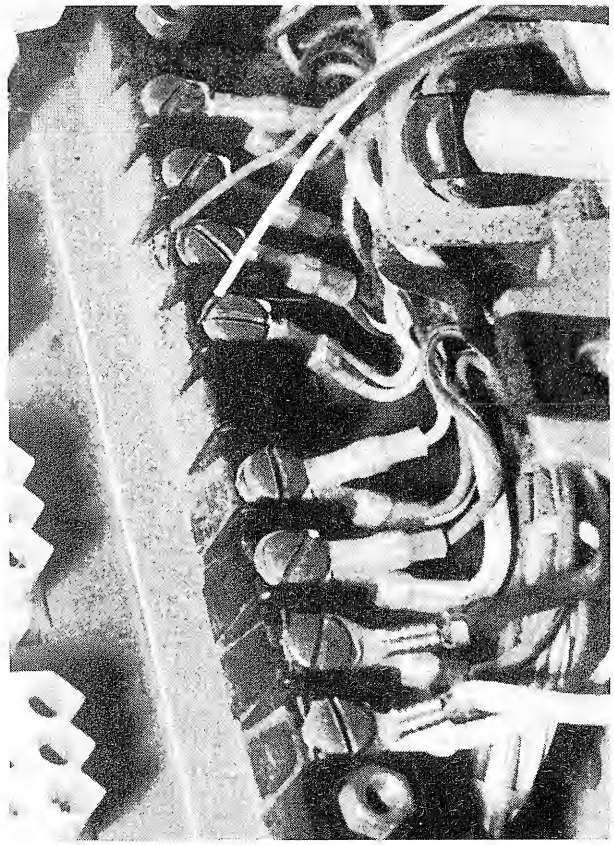
```
10 CLS:CLEAR 1000 :DIM A$(55)
20 FOR L=1 TO 55
30 INPUT A$(L): IF LEN(A$(L))>64 PRINT"LINE TOO LONG": GOTO 30
40 IF A$(L)="" THEN GOTO 100 ELSE NEXT L
100 LPRINT" ":LPRINT" "
110 FOR L=1 TO 55
120 LPRINT A$(L): IF A$(L)="" THEN GOTO 200 ELSE NEXT L
200 LPRINT" ":LPRINT" ":LPRINT" ":CLS
210 INPUT"DO YOU WANT ANOTHER COPY OF THE SAME PRINTOUT (Y/N)"
220 Q$=INKEY$:IF INKEY$="" GOTO 220
230 IF Q$="Y" GOTO 100 ELSE GOTO 10
240 END
```

When you run the above program, you will be presented with just a blank screen on your '80, with the cursor showing. You can type in up to 55 lines of text (each line cannot be longer than 64 characters), so you must remember to hit ENTER or NEW LINE at the end of each line. To obtain a printout, make sure your printer is connected and ready to go, then hit ENTER or NEW LINE again after the last line of text. The computer will respond to the input of a null string and begin printing your text.

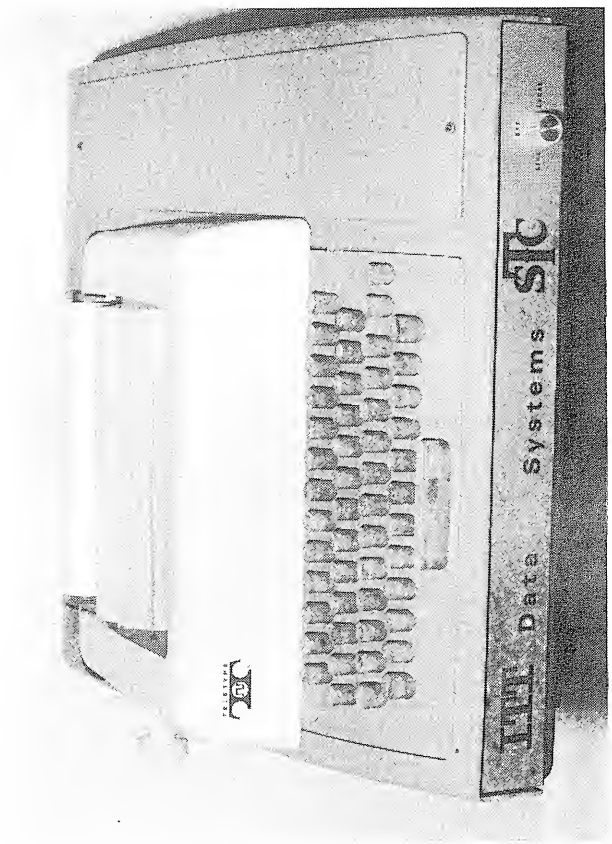
To use the simple interface with the Electric Pencil word processing program (tape or disk versions) go into the COMMAND mode using the SHIFT K option and set the Baud Rate to 110, Line Length to 72 and the number of nulls on each carriage-return to 2. If you want your text to be right-justified, you can also type in J1. (It is turned off again with J0, which is the default value).



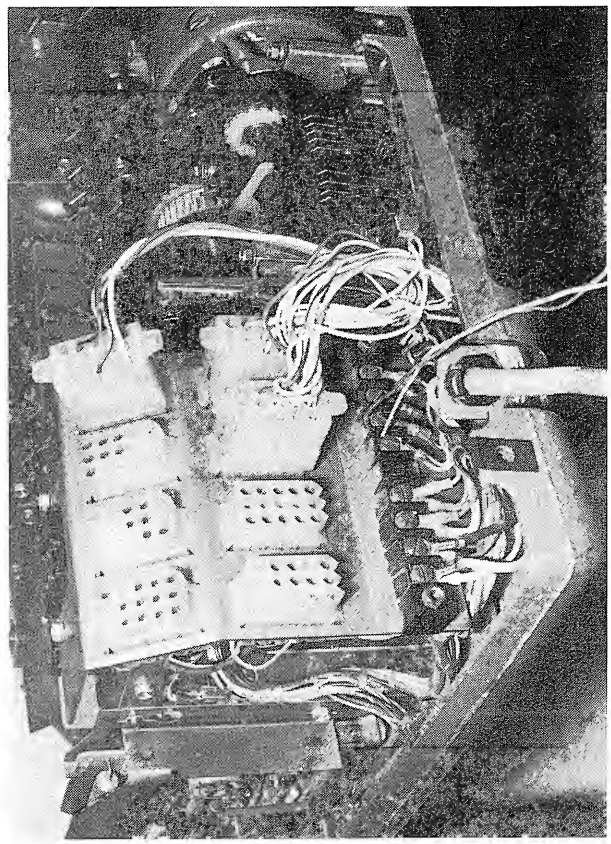
VIEW OF INTERFACE AND BATTERY



CLOSE-UP OF TERMINAL BLOCK



MODEL 33 TELETYPE



VIEW OF MODEL 33 WITH COVER REMOVED

Type in the following:

B1 ENTER
L72 ENTER
F2 ENTER

For the right-justification option, you may, at your option, type in:

J1 ENTER

Return to the text-entry mode by pressing SHIFT S.

You are then ready to enter and edit text and can print out your hard copy by setting the cursor to the beginning of the text and hitting SHIFT P. Be sure the teletype is switched on to its LINE position and the el-cheapo interface is connected to the battery terminals and the TRS-80's AUX grey plug before you enter the SHIFT P command and watch your teletype do its stuff.

If your TRS-80 is fitted with an expansion interface, you will not be able to use the el-cheapo interface with Electric Pencil. You will just have to instal a proper RS.232C serial communications port and do your printouts using the UART option (by entering the COMMAND mode as above, and hitting U ENTER before the other specifications for the printer-driver control).

Last but not least, the trusty old Model 33 may be slow and noisy and only print in upper case but that all-caps printout can actually be an advantage when LLISTing your programs. The printout is better than most dot-matrix printers costing twice the price and can closely rival the printouts from an electric typewriter. Just be sure to disconnect the crocodile clips from the lantern battery when you're not using the interface, or even at 20 milli-amps, you'll eventually drain it of power.

SEE PAGE 8 FOR CIRCUIT DIAGRAM

- 0000000000 -

***** THE THEORY AND TECHNIQUES OF SORTING - Part 5 by B. Simson *****

This month, I shall demonstrate a method of sorting by selection. However, this is not your common garden variety "selection sort" as discussed in earlier articles. This one will use a sophisticated data structure, known as..... THE TREE (special garden variety!). This is sounding more like a nature documentary than a discussion on computing techniques!

TREE DATA STRUCTURES

A tree is a data structure that uses pointers,. similar to linked linear lists, but instead of just one pointer to the successor node, a tree consists of nodes that have multiple pointers to multiple successor nodes. While this may seem complicated to visualize, the type of tree that is used in this tree sort is a simple version of a general tree, being a binary tree. A binary tree is a tree where the maximum number of successor nodes is two. Therefore, each node consists of just two pointers, and can be conceptually illustrated as follows...

LPTR and RPTR represent the left and right pointers of the node to successor nodes.

INFO represents the actual data contained in that node, which is irrelevant to the actual data structure itself.

Figure 1 illustrates a binary tree data structure at a conceptual level. It is easy to see why such a structure is called a tree...bit like a family tree. The first node (one at the top) is known as the root and the nodes that do not have any successor nodes are known as (you guessed it)....leaves. Therefore, by definition, leaf nodes have both pointers set to null.

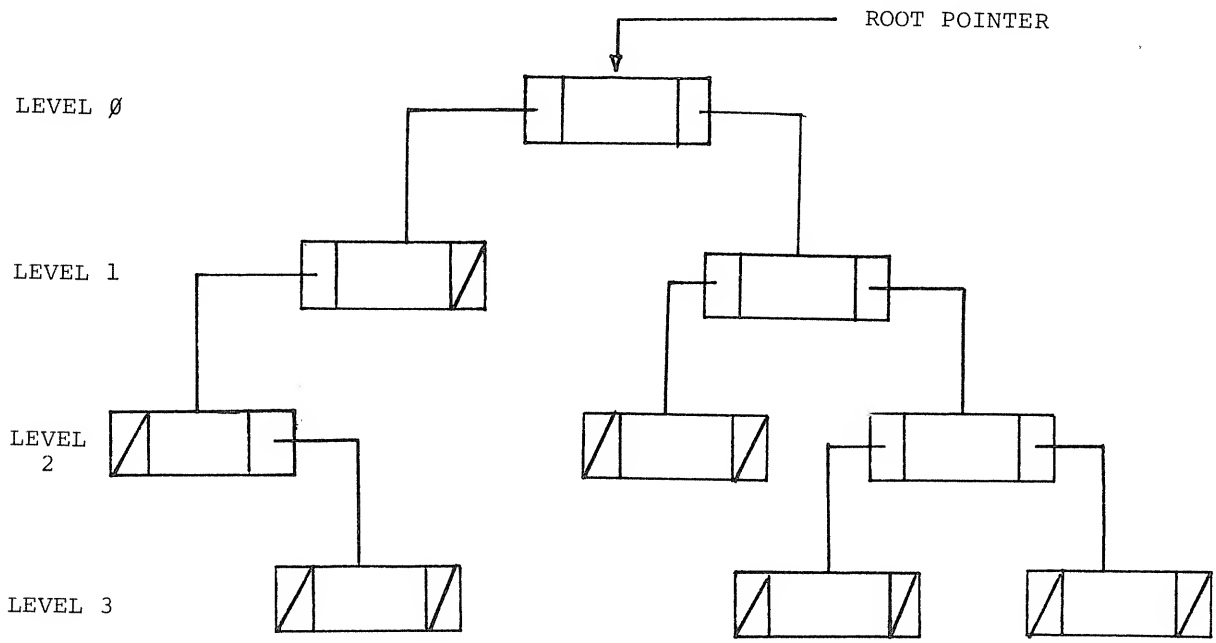
TREE SORTING

Well you may think that all this is a nice bit of theory, but what does it have to do with sorting? Items can be efficiently sorted if they are structured into a binary tree, then "selected" (hence sorting by selection) from the tree (rather like picking fruit!) using a technique known as tree traversal. There are several traversal methods available to binary trees. The one applicable to tree sorting is the "inorder traversal" method. There are also many techniques of sorting using tree data structures. For the purposes of this discussion, the method outlined is simply termed the "tree sort". This tree sort involves two phases:

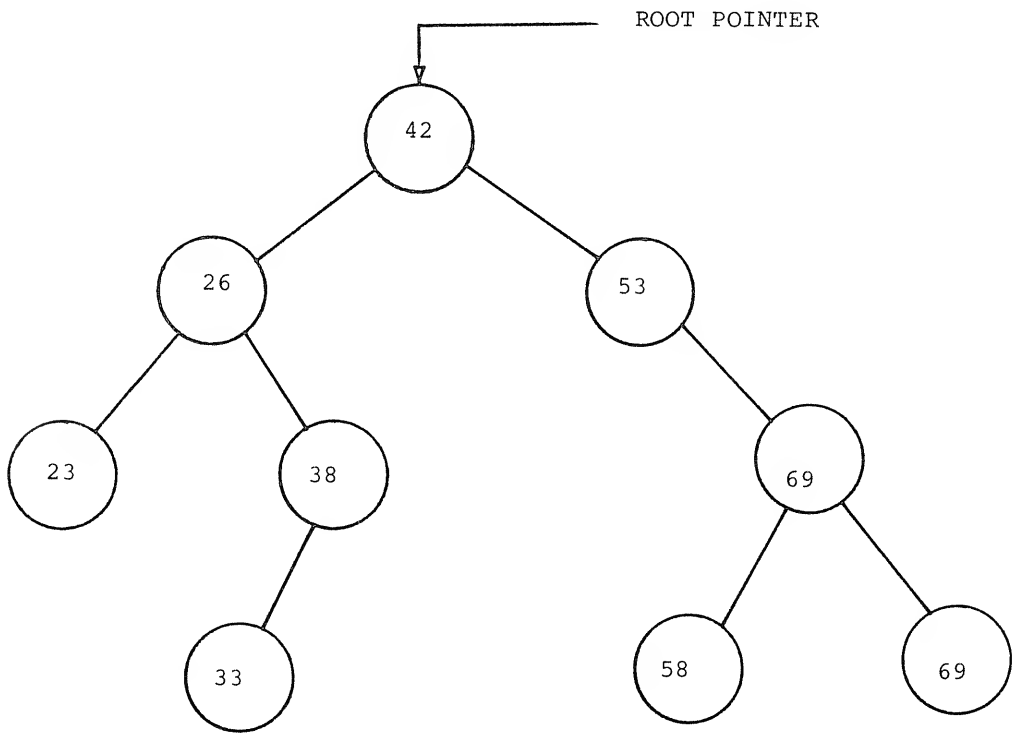
1. Building a binary tree structure of the data
2. Traversing the tree using the inorder method.

BINARY TREE CREATION

Tree data structures cannot be directly represented as a dynamic data structure by BASIC. Instead, they must be simulated using arrays. Therefore, the unsorted data must be placed into a 2 dimensional array of N rows and 3 columns, where N is the number of items being sorted, column 1 represents the Left pointer of the node, column 2 the Data, column 3 the Right pointer. Alternatively, the data can be placed in a conventional one-dimensional array, with a further 2-dimensional array, or 2 single-dimension arrays, for the Left and Right node pointers. This discussion will adopt the former approach.



(a) Conceptual view of binary tree structure



(b) Binary tree with sample data

Figure 1 - Two views of binary tree structures

The tree building algorithm is shown in Figure 2. This algorithm does not move the items for sorting (column 2). Instead, a link structure is built around the data using columns 1 and 3 of the array. Each candidate node for linking to the tree (indexed by I) is compared with nodes already linked to the tree by chaining through the tree. If the new node has an INFO value that is less than the current node being examined (indexed by P), then the left subtree

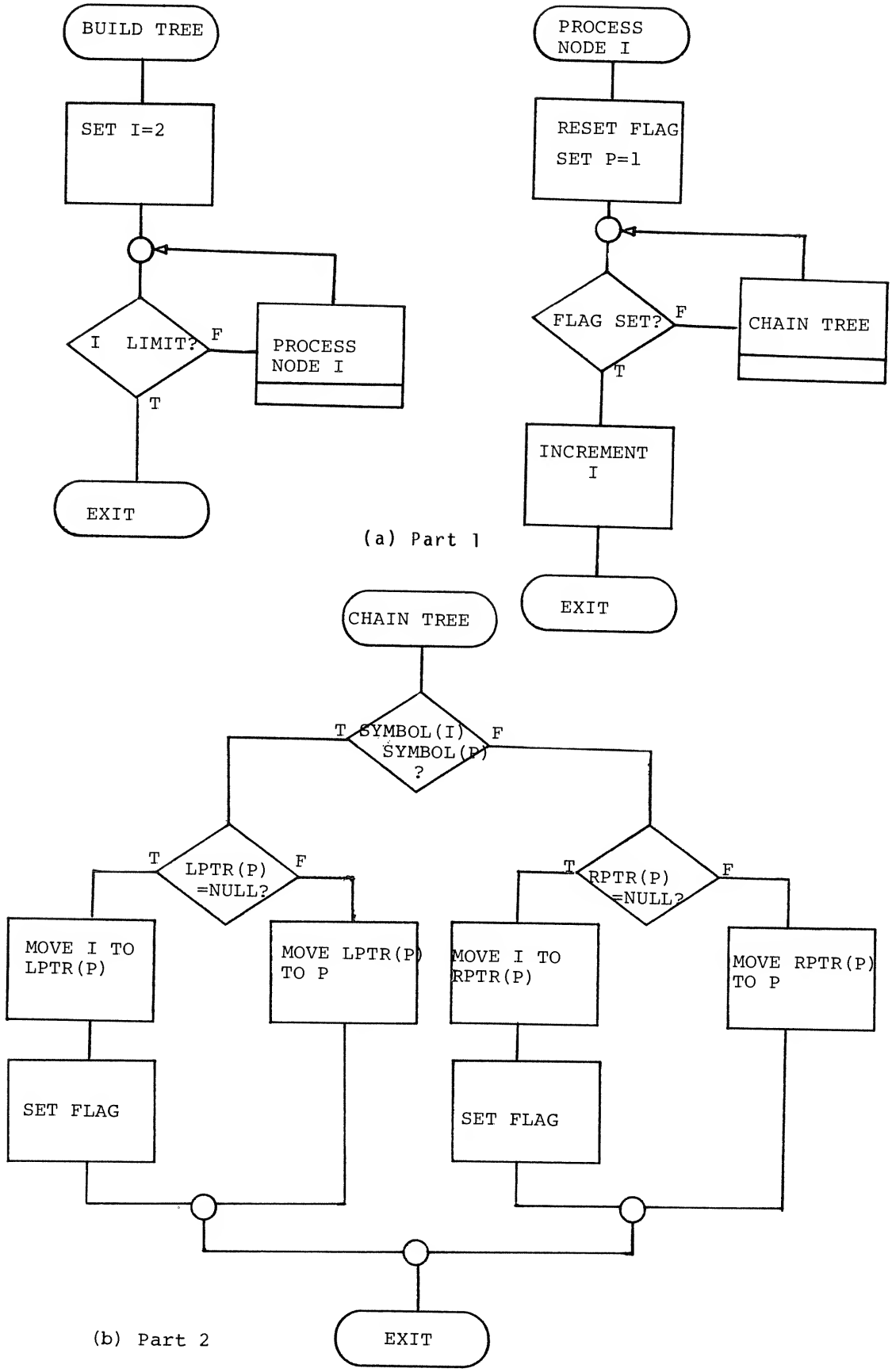


FIGURE 2 - Tree creation phase algorithm

I	Address of node to be linked
P	Address of current node in tree
LIMIT	Size of input data
FLAG	Set when node at I linked

is searched, otherwise the right subtree is searched. This means that when the structure is completely built, each non-leaf node will have a successor node of less value on its left, and a successor node of equal or greater value on its right. An example of the relationship between the data is shown in Figure 1 (B).

Assuming the stage of the tree being built is as Figure 1 (B), and the next node to be linked to the tree has a value of 40, the process of searching subtrees in order to link this node will be as follows:

40 is less than 42 (root), so subtree 26 is searched.
40 is greater than 26, so subtree 38 is searched.
40 is greater than 38, and the RPTR to node 38 is null, therefore, node 40 is linked to the right position of node 38, updating the RPTR of node 38.

When the tree structure is completely built, the data will be in the same position as before, but columns 1 and 3 of the tree simulation array (LPTR and RPTR) will contain addresses to other parts of the array to form a network of relationships as illustrated in Figure 1 (B).

The concept of subtrees is an important one. The tree in Figure 1 (B) can be visualized as being composed of 2 subtrees whose roots have values 26 and 53. In turn, "tree 26" can be visualized as being composed of two subtrees whose roots have values 23 and 38 and so on. The subtree definition ends with leaf nodes. Tree 23 is composed of just a root, whereas tree 38 has just one leaf node. Subtree analysis is also used in the traversal process.

Using Figure 2, the BASIC code can be written to build the binary tree, using the input data in the format described above (i.e. in column 2 of an (N by 3) array) residing in array "T". The tree building module is as follows:

```
910 FORI=2TOLM
920 LINKED=FALSE:P=1
930 IF LINKED THEN 960
940 IFT(I,2)<T(P,2)THEN IFT(P,1)=OTHENT(P,1)=I:LINKED=TRUE ELSEP=T(P,1) ELSE IFT
(P,3)=OTHENT(P,3)=I:LINKED=TRUE ELSEP=T(P,3)
950 GOTO930
960 NEXT
962 RETURN
```

Once the tree has been completely built (or grown), its fruit is now ripe for picking. This involves "traversing" the tree in a prescribed manner, visiting every node in the tree (picking all its fruit!).

INORDER TRAVERSAL

There is more than one way in which all the nodes in the tree can be processed in the INORDER fashion. One way is to use a recursive algorithm, which results in fairly compact and easily abstracted code. Recursively....the recursive definition of INORDER traversal is:

traverse the Left subtree using INORDER method,
process the Root node,
traverse the Right subtree using INORDER method.

As can be seen, this procedure is defined in terms of itself, a feature of recursion. Also, it can easily be seen that the procedure views the tree as consisting of subtrees, which themselves consist of further subtrees, etc. This means that a particular root node of a subtree cannot be processed until the left subtree to that node is completely processed in the same manner. If there is no left subtree (LPTR=Null) then that imaginary left subtree is deemed to be completely processed. After the root node is then processed, the right subtree to that node is then completely processed in the same manner.

Iteratively....Since recursion is not supported in all languages, an alternative algorithm for the more common iterative approach is shown in Figure 3.

Since it is required to descend and subsequently ascend parts of the tree during the traversal process, pointers which will allow movement up the tree must be temporarily stored. The information that already exists in the structure of the tree allows downward movement from the root but, because movement up the tree must be made in the reverse manner, a stack is required to store pointers as the tree is traversed. The stack data structure provides the last-in-first-out feature required for this operation. The box "process P" in Figure 3 actually deals with a node, whereas the other parts of the algorithm deal with movement up or down the tree. It is at this point that the "fruit" from the tree is "picked", by outputting the INFO value of that node to an output list, being the result list after sorting. The module for the INORDER traversal of the binary tree is shown below. The following variables used are explained:

S = Array simulating stack, SP = stack pointer
OL = output list, OL = current index to that list.

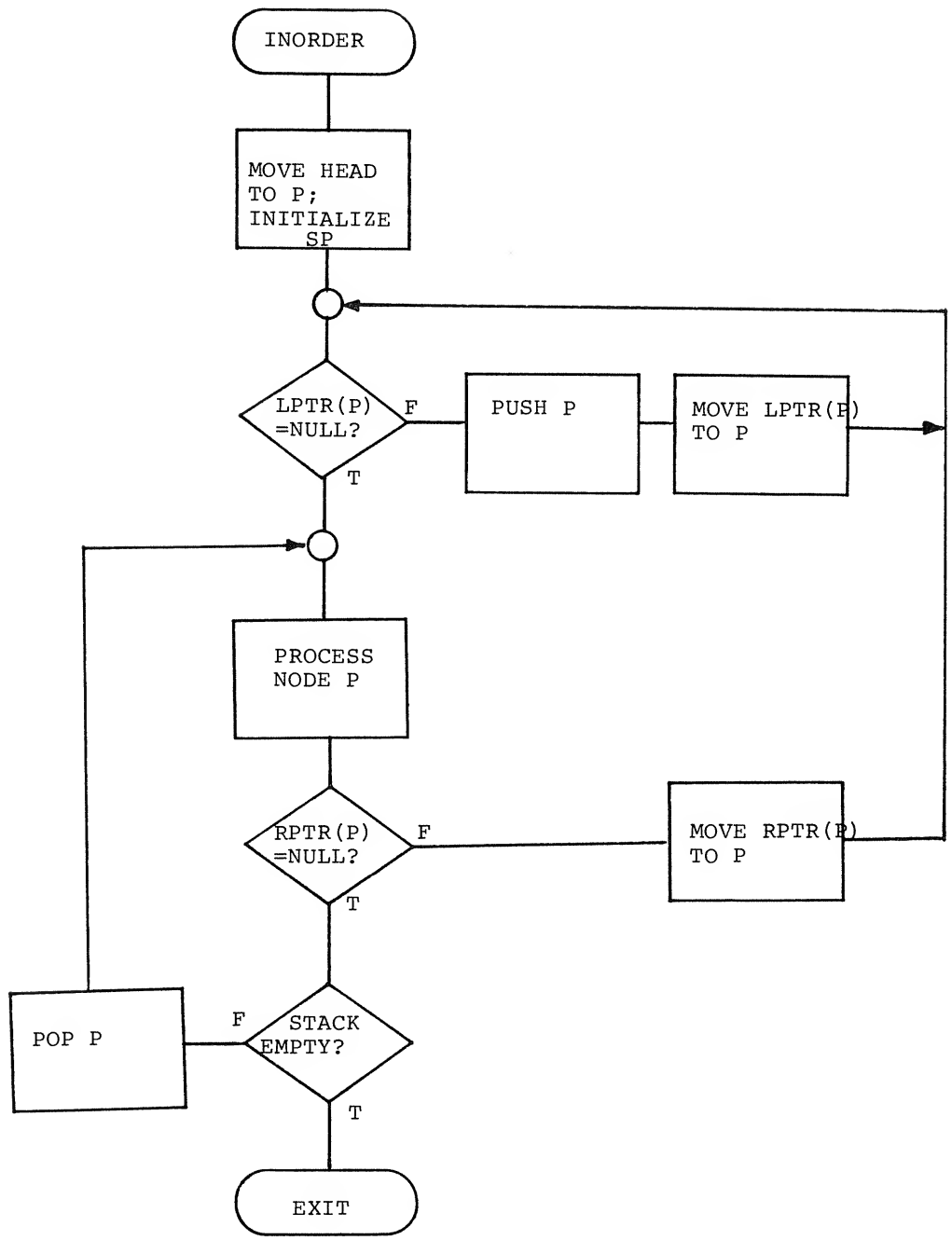


FIGURE 3 - Inorder traversal of binary tree (iterative method)

HEAD Address of root of tree
P Address of current node
SP Stack pointer

```
1580 '    INORDER TRAVERSAL OF BINARY TREE
1590 P=1:SP=0:OP=0
1600 IFT(P,1)<>0THENSP=SP+1:S(SP)=P:P=T(P,1):GOTO1600
1610 OP=OP+1:OL(OP)=T(P,2)
1620 IFT(P,3)<>0THENP=T(P,3):GOTO1600
1630 IFSP=0GOTO1660
1640 P=S(SP):SP=SP-1
1650 GOTO1610
1660 RETURN
```

Line 1600 checks for the presence of a Left subtree. If one exists, the current root node address is stacked and the Left subtree is traversed in the same manner.

Line 1610 outputs the current root node after the Left subtree has been processed.

DON'T BE HELD BACK BY AN ANTIQUATED DISK OPERATING SYSTEM

MOVE UP TO

NEWDOS 80 **$149 incl. p&p**

NEWDOS 80 is a completely new DOS for the TRS-80 SYSTEM 80. It is well-documented, bug free and increases the power of your system many times over. It is upward compatible with TRSDOS AND NEWDOS (ie TRSDOS and NEWDOS+ programs will run on NEWDOS 80 but the reverse is not necessarily so).

These are just a few of the many new features offered by NEWDOS 80.

* New BASIC commands that support variable record lengths up to 4095 bytes long.
* Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5¼ inch mini disk drives, 8 inch disk drives OR ANY COMBINATION.
* An optional security boot-up for BASIC or machine code application programs. User never sees "DOS-READY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements, including "LIST".
* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
* Enhanced and improved RENUMBER that allows relocation of subroutines.
* Create powerful chain command files which will control the operation of your system.
* Device handling for routing to display and printer simultaneously.
* MINIDOS – striking the D, F and G keys simultaneously calls up a MINIDOS which allows you to perform many of the DOS commands without disturbing the resident program.
* Includes Superzap 3.0 which enables you to display/print/modify any byte in memory or on disk.
* Also includes the following utilities:
  - Disk Editor/Assembler
  - Disassembler (Z80 machine code)
  - LM offset – allows transfers of any system tape to Disk file – automatically relocated.
  - LEVEL I – Lets you convert your computer back to Level 1.
  - LVIDKSL – Saves and loads Level 1 programs to disk.
  - DIRCHECK – Tests disk directories for errors and lists them.
  - ASPOOL – An automatic spooler which routes a disk file to the printer whilst the computer continues to operate on other programs.
  - LCDVR – a lower case drives which display lower case on the screen if you have fitted a simple lower case modification.

DISK DRIVE USERS ELIMINATE CRC ERRORS AND TRACK LOCKED OUT MESSAGES FIT A PERCOM DATA SEPARATOR

$37.00 plus $1.20 p&p.

When Tandy designed the TRS-80 expansion interface, they did not include a data separator in the disk-controller circuitry, despite the I.C. manufacturer's recommendations to do so. The result is that many disk drive owners suffer a lot of Disk I/O errors. The answer is a data separator. This unit fits inside your expansion interface. It is supplied with full instructions and is a must for the serious disk user.

MPI DISK DRIVES HIGHER PERFORMANCE – LOWER PRICE

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as 8" drives and consequently, they have the fastest track-to-track access time available – 5msec! All MPI drives are capable of single or double-density operation. Double-density operation requires the installation of a PERCOM doubler board in the expansion interface.

As well as single head drives, MPI also makes dual-head drives. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare or in a metal cabinet – set up to operate with your TRS-80 or SYSTEM 80. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

MPI B51 40 Track Single Head Drive. . . . . only $349
MPI B52 40 Track Double Head Drive. . . . . only $449

Prices are for bare drives and include p&p. Add $10.00 per drive for a cabinet and $60.00 for a power supply to suit two drives. 40 track drives are entirely compatible with 35 track drives. A 40 track DOS such as NEWDOS 80 is necessary to utilise the extra 5 tracks.

OVER 800 KILOBYTES ON ONE DISKETTE! WITH MPI 80 TRACK DRIVES

MPI 80 track drives are now available. The B91 80 track single-head drive stores 204 Kilobytes of formatted data on one side of a 5¼ inch diskette in single-density mode. In double-density mode it stores 408 Kilobytes and loads/saves data twice as quickly.

The B92 80 track dual-head drive stores 204 Kilobytes of formatted data on EACH side of a 5¼ inch diskette in single-density mode. That's 408 Kilobytes per diskette. In double-density mode, the B92 stores a mammoth 408 Kilobytes per side or 816 Kilobytes of formatted data per diskette. With two B92's and a PERCOM double, you could have over 1.6 Megabytes of on line storage for your TRS-80 for less than $1500!!

MPI B91 80 Track Single Head Drive. . . . . only $499
MPI B92 80 Track Dual Head Drive . . . . . only $619

Prices are for bare drives and include p&p. Add $10.00 per drive for a cabinet and $60.00 for a power supply to suit two drives. Note: 80 track drives will not read diskettes written on a 35 or 40 track drive. If drives with different track counts are to be operated on the same system, NEWDOS 80 must be used.

CARE FOR YOUR DISK DRIVES? THEN USE 3M's DISK DRIVE HEAD CLEANING DISKETTES

$30.20 incl. p&p.

Disk drives are expensive and so are diskettes. As with any magnetic recording device, a disk drive works better and lasts longer if the head is cleaned regularly. In the past, the problem has been, how do you clean the head without pulling the mechanism apart and running the risk of damaging delicate parts. 3M's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

- 2 head cleaning diskettes
- 1 bottle of cleaning fluid
- 1 bottle dispenser cap

PLEASE USE ORDER FORM ON PAGE 36

USE TANDY PERIPHERALS ON YOUR SYSTEM-80 VIA SYSPAND-80 - $97.50 incl. p&p

The SYSTEM-80 hardware is not compatible with the TRS-80 in two important areas. The printer port is addressed differently and the expansion bus is entirely different. This means that SYSTEM-80 owners are denied the wealth of economical, high performance peripherals which have been developed for the TRS-80. Until now, that is. MICRO-80 has developed the SYSPAND-80 adaptor to overcome this problem. A completely self-contained unit in a small cabinet which matches the colour scheme of your computer, it connects to the 50-way expansion part on the rear of your SYSTEM 80 and generates the FULL Tandy 40 way bus as well as providing a Centronics parallel printer port. SYSPAND-80 enables you to run an Exatron Stringy Floppy from your SYSTEM 80, or an LNW Research expansion interface or any other desirable peripherals designed to interface to the TRS-80 expansion port. Make your SYSTEM 80 hardware compatible with the TRS-80 via SYSPAND-80.

PROGRAMS BY MICROSOFT

EDITOR ASSEMBLER PLUS (L2/16K) $37.50 + $1.20 p&p

A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

LEVEL III BASIC $59.95 plus $1.20 p&p

Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

ADVENTURE ON DISK $35.95 plus $1.20 p&p

This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

BASIC COMPILER $208 plus $2.00 p&p

New improved version, the Basic Compiler converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate.

UPGRADE TO 16K FOR ONLY $30.00!!

MICRO-80's 16K MEMORY EXPANSION KIT HAS BEEN REDUCED IN PRICE EVEN MORE

Larger volume means we buy better and we pass the savings on to you. These are our proven, prime, branded 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 — these cost an additional $4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

DISK DRIVE CABLES SUITABLE FOR ANY DISK DRIVES

DC-2 2 Drive Connector Cable . . . . . $39 incl. p&p
DC-4 4 Drive Connector Cable . . . . . $49 incl. p&p

DOUBLE THE SPEED AND CAPACITY OF YOUR DISK DRIVES PERCOM DOUBLER ONLY $220 plus $2.00 p&p

Installing a Doubler is like buying another set of disk drives, only much cheaper!! The doubler works with most modern disk drives including:- MPI, Micropolis, Pertec, TEAC (as supplied by Tandy). The doubler installs in the TRS-80 expansion interface, the System-80 expansion interface and the LNW Research expansion interface in a few minutes without any soldering, cutting of tracks, etc. It comes complete with its own TRSDOS compatible double density operating system.

DOUBLE-ZAP II – DOUBLE DENSITY PATCH FOR NEWDOS 80 ONLY $53.00 plus $1.00 p&p

DISCONTINUED

If you are using NEWDOS 80, then you also need DOUBLE-ZAP II on diskette. This program upgrades your NEWDOS 80 to double density with ADR (automatic density recognition). It retains all the familiar features, including the ability to mix and match track counts on the same cable. In addition, it gives NEWDOS 80 the ability to mix densities on the same cable, automatically. If you place a single density diskette in drive Ø, say and a double density diskette in drive 1, Double-ZapII will recognise this and read/write to drive Ø in single density whilst at the same time it reads/writes to drive 1 in double density!

FLOPPY DOCTOR AND MEMORY DIAGNOSTIC (by MICRO CLINIC) $29.95 plus 50c. p&p

Two machine language programs on a diskette together with manual which thoroughly test your disk drives and memory. There are 19 possible error messages in the disk drive test and their likely causes are explained in the manual. Each pass of the memory tests checks every address in RAM 520 times, including the space normally occupied by the diagnostic program itself. When an error occurs the address, expected data, and actual data are printed out together with a detailed error analysis showing the failing bit or bits, the corresponding IC's and their location. This is the most thorough test routine available for TRS-80 disk users.

BOOKS

LEVEL II ROM REFERENCE MANUAL $24.95 + $1.20 p&p

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES $24.95 + $1.20 p&p

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories — this is a must for the serious Disk user and is a perfect companion to any of the NEWDOS's.

LEARNING LEVEL II $16.95 + $1.20 p&p

Written by Daniel Lien, the author of the TRS-80 Level I Handbook, this book teaches you, step-by-step, how to get the most from your Level II machine. Invaluable supplement to either the TRS-80 Level II Manual or the System-80 Manuals.

MORE AUSTRALIAN SOFTWARE

All programs designed to run on both the TRS-80 or the SYSTEM 80 without modification. Most programs include sound

TRIAD VOL 1 – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Three separate games which test your powers of memory and concentration. The programs combine graphic displays and sound:

SIMON-SEZ: Just like the electronic music puzzles on sale for more than $20. Numbers are flashed on the screen and sounded in a sequence determined by the computer. Your task is to reproduce the sequence, correctly.

LINE?: Rather like a super, complicated version of noughts and crosses. You may play against another player or against the computer itself. But beware, the computer cheats!

SUPER CONCENTRATION: Just like the card game but with more options. You must find the hidden pairs. You may play against other people, play against the computer, play on your own, or even let the '80 play on its own.

TRIAD VOL 2 – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Remember those "NUMERO" puzzles in which you had a matrix of numbers (or letters) with one blank space and you had to shuffle the numbers around one at a time until you had made a particular pattern? Well, **SHUFFLEBOARD**, the first program in this triad, is just this, except that the computer counts the number of moves you take to match the pattern it has generated – so it is not possible to cheat.

MIMIC is just like SHUFFLEBOARD except that you only see the computer's pattern for a brief span at the beginning of the game, then you must remember it!

In **MATCHEM**, you have to manoeuvre 20 pegs from the centre of the screen to their respective holes in the top or bottom rows. Your score is determined by the time taken to select a peg, the route taken from the centre of the screen to the hole and your ability to direct the peg into the hole without hitting any other peg or the boundary.

VISURAMA L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Two programs which give fascinating, ever-changing patterns on the screen.

LIFE is the fastest implementation of the Game of Life you will see on your '80. Machine language routines create up to 1200 new generations per minute for small patterns or up to 100 per minute for the full 128 x 48 screen matrix. Features full horizontal and vertical wraparound.

EPICYCLES will fascinate you for hours. The ever-changing ever-moving patterns give a 3D effect and were inspired by the ancient Greek theories of Ptolemy and his model of the Solar system.

EDUCATION AND FUN – L1/4K, L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Written by a primary school teacher to make learning enjoyable for his pupils, there are five programs in both Level I and Level II to suit all systems:

BUG-A-LUG: a mathematics game, in which you must get the sum correct before you can move.

AUSTRALIAN GEOGRAPHY: learn about Australian States and towns, etc.

SUBTRACTION GAME: build a tower with correct answers.

HOW GOOD IS YOUR MATHS? Select the function (+, –, ÷ or X) and degree of difficulty.

HANGMAN: That well known word game now on your computer.

Recommended for children from 6 to 9 years.

COSMIC FIGHTER & SPACE JUNK – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Both programs have sound to complement their excellent graphics. In **COSMIC FIGHTER**, you must defend the earth against seven different types of alien aircraft. It is unlikely that you will be successful but you will have a lot of fun trying!

You mission in **SPACE JUNK** is to clean up all the debris left floating around in space by those other space games. It is not as simple as it sounds and space junk can be quite dangerous unless you are very careful.

SPACE DRIVE L2/4K & 16K
Cassette $8.95 Disk $13.95
+ 60c p&p

Try to manoeuvre your space ship through the meteor storms then land it carefully at the space port without running out of fuel or crashing. Complete with realistic graphics.

STARFIRE AND NOVA INVASION L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Both programs include sound to improve their realism.

STARFIRE seats you in the cockpit of an X-wing fighter as you engage in battle with the deadly Darth Vader's Tie-fighters. Beware of the evil one himself and may the Force be with you.

In **NOVA INVASION**, you must protect your home planet of Hiberna from the invading NOVADIANS. You have two fixed guns at each side of the screen and a moveable one at the bottom. Apart from shooting down as many invaders as possible, you must protect your precious hoard of Vitaminium or perish!

AIR ATTACK AND NAG RACE – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

An unlikely combination of programs but they share the same author who has a keen sense of humour.

AIR ATTACK includes sound and realistic graphics. The aircraft even have rotating propellors! But they also drop bombs on you, so it's kill or be killed!

NAG RACE lets you pander to your gambling instinct without actually losing real money. Up to five punters can join in the fun. Each race results in a photo-finish whilst there is a visible race commentary at the bottom of the screen throughout the race. Happy punting!

FOUR LETTER MASTERMIND L2/16K
Cassette $8.95 Disk $13.95
+ 60c p&p

There are 550 four-letter words from which the computer can make its choice. You have 12 chances to enter the correct word. After each try, the computer informs you of the number of correct letters and those in the correct position. You can peek at the list of possible words but it will cost you points. Makes learning to spell fun.

MUSIC IV – L2/16K
Cassette $8.95 Disk $13.95
+ 60c p&p

Music IV is a music compiler for your '80. It allows you to compose or reproduce music with your computer that will surprise you with its range and quality. You have control over duration (full beat to 1/16 beat) with modifications to extend the duration by half or one third for triplets. Both sharps and flats are catered for as are rests. Notes on whole sections may be repeated. The program comes with sample data for a well-known tune to illustrate how it is done.

*****SAVE 00$'s ***SAVE 00$'s ***SAVE 00$'s ***MICRO-80 EXPANSION INTERFACE *****

MICRO-80's expansion interface utilises the proven LNW Research Expansion board. It is supplied fully built up and tested in an attractive cabinet with a self contained power supply, ready to plug in and go. The expansion interface carries MICRO-80's full, no hassle, 90-day warranty.

Features include:- • Sockets for up to 32K of memory expansion • Disk controller for up to 4 disk drives • Parallel printer port • Serial RS232C/20mA I/O port • Second cassette (optional)

The expansion interface connects directly to your TRS-80 L2/16K keyboard or, via SYSPAND-80 to your SYSTEM 80VIDEO GENIE

Prices: HD-010-A Expansion Interfaces with Ø K : $499.00 HD-010-B Expansion Interfaces with 32K : $549.00 HD-011 Data separator fitted (recommended) : add $29.00 HD-012 Dual cassette Interfaces fitted : add $19.00

The MICRO-80 Expansion Interface is also available in kit form.

Prices: HD-013 Kit consisting of LNW Research PC board and manual, ALL components including cabinet & power supply : $375.00 HD-011 Data separator for above $25.00 HD-013 Dual cassette Interface kit : $15.00

TURN THIS into this

for $49.00 plus $2.00 p & p

A choice of upper and lower case display is easier to read, gives greater versatility.

The Micro-80 lower case modification gives you this facility, plus the symbols for the 4 playing-card suits for $49.00 + $2.00 p. & p.

The Micro-80 modification features true below-the-line descenders and a block cursor.

Each kit comes with comprehensive fitting instructions and two universal lower-case drive routines on cassette to enable you to display lower case in BASIC programs.

The driver routines are self-relocating, self-protecting and will co-reside with other machine language programs such as Keyboard-debounce, serial interface driver programs etc.

Both programs give your TRS-80™ Model I or System 80™ an optional typewriter capability, i.e. shift for upper case

The second programme also includes Keyboard-debounce and a flashing cursor.

You fit it. Or we can.

Fitting the modification requires soldering inside the computer. This should only be carried out by an experienced hobbyist or technician.

If you are at all dubious, a fitting service is available in all capital cities for only $20.00.

A list of installers is included with each kit.

Save $120 now.

ADD A DISK DRIVE TO YOUR TRS-80™ MODEL III FOR ONLY $875.00 OR ADD TWO FOR ONLY $1199.

The Micro-80 disk drive upgrade for the TRS-80™ Model III contains the following high quality components:

1 or 2 MPI 40-track single head disk drives, 1 VR Data double-density disk controller board and 1 dual drive power supply plus all the necessary mounting hardware, cables and comprehensive fitting instructions, which can be carried out with a minimum of fuss by any average computer owner.

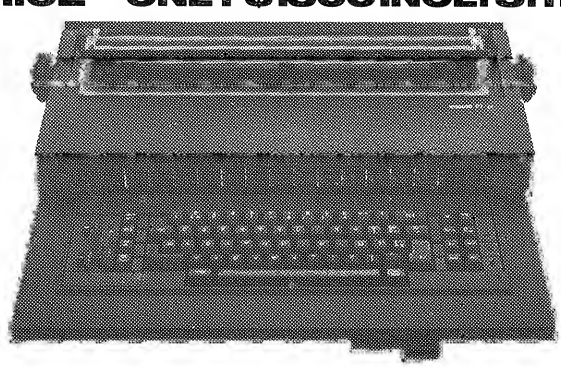
Fitting service is available for $25.00 in most capital cities.

SPECIAL SALE PRICE—ONLY $1500 INCL. S.T.

Daisy Wheel Typewriter/Printer

MICRO-80 has converted the new OLIVETTI ET-121 DAISY WHEEL typewriter to work with the TRS-80 and SYSTEM 80 or any other microcomputer with a Centronics parallel port (RS 232 serial interface available shortly). The ET-121 typewriter is renowned for its high quality, fast speed (17 c.p.s.), quietness and reliability. MICRO-80 is renowned for its knowledge of the TRS-80/SYSTEM 80 and its sensible pricing policy. Together, we have produced a dual-purpose machine:- an attractive, modern, correcting typewriter which doubles as a correspondence quality Daisy-wheel printer when used with your micro-computer.

How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.



BUY YOUR MODEL 3 FROM MICRO-80 AND SAVE $000's

MICRO-80 fits reliable MPI disk drives to the TRS-80 Model 3 to give system capacities and capabilities far in excess of those available elsewhere. All our conversions utilise low dissipation, switching-mode power supplies to avoid screen jitter and overheating. The disk controller boards used incorporate special compensation circuitry for 80 track disk drives and may also be used to run 8 inch disk drives with an appropriate cable and DOS.

MODEL 340
2 40 TRACK SINGLE-HEAD DISK DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM **$3130**

MODEL 340+
2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM **$3350**

MODEL 380+
2 80 TRACK DUAL-HEAD DRIVES GIVING
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM **$3800**

★ *NEW* ★ ★ *NEW* ★ ★ *NEW* ★

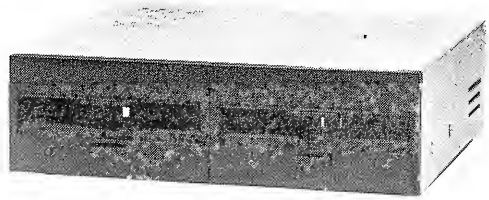
MODEL 500 — 5+ MEGABYTE MODEL 3
1 40 TRACK DUAL-HEAD DRIVE GIVING
350K OF FLOPPY DISK STORAGE FOR TRANSFERRING
PROGRAMS AND BACKUP, 48K RAM, EXTERNAL
5 MEGABYTE WINCHESTER SUB-SYSTEM,
CP/M (ORG 4200N) DISK OPERATING SYSTEM **$5895**

The MODEL 500 offers the high speed, mass storage capacity and reliability of a Winchester drive for thousands of dollars less than you would pay for any comparable system. Model 500 is a serious business computer able to tackle the most demanding tasks.

All prices are in Australian dollars, include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All computers and systems carry MICRO-80's 90-day Warranty covering parts and labour.

SAVE A PACKET ON MICRO-80's DISK DRIVE PACKAGES FOR TRS-80 MODEL 1 AND SYSTEM 80 MICROCOMPUTERS

SINGLE DRIVE PACKAGE *from* ... $499



DUAL DRIVE PACKAGE *from* ... $874

Bigger volume means lower cost price, which we are passing on to you. Avoid the annoying bundle of cables, wires and separate boxes. MICRO-80 is now offering our well-proven MPI disk drives in attractive, self-contained single or dual-drive cabinets complete with internal power supply. Our drive Ø and dual-drive packages also include the appropriate version of DOSPLUS and dual-drive cable.

The best news of all is the specially reduced package prices ...
SAVE $23 — $107 over our already low prices!

Choose the appropriate system from the table below:

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price	* Saving
DRIVE Ø						
1 x MPI B51	40	1	100K	3.3	$499	$77.95
1 x MPI B52	40	2	200K	3.4	$639	$97.95
1 x MPI B92	80	2	400K	3.4	$799	$107.95
DRIVE 1						
1 x MPI B51	40	1	100K	—	$415	$23.00
1 x MPI B52	40	2	200K	—	$525	$23.00
1 x MPI B92	80	2	400K	—	$695	$23.00

*Represents the saving compared with buying all the items included in the package separately

•Drive Ø package includes one bare disk drive, self-contained single-drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.

•Drive 1 package includes one bare disk drive and self-contained single-drive cabinet/power supply as illustrated.

If it's a dual-drive system you need, then take advantage of our dual-drive package and
SAVE a further $40 on the price of two single-drive packages ...

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price
2 x MPI B51	40 ea	1 ea	2 x 100K	3.3	$874
2 x MPI B52	40 ea	2 ea	2 x 200K	3.4	$1125
2 x MPI B92	80 ea	2 ea	2 x 400K	3.4	$1454

Dual-drive package includes two bare disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of Dosplus indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

All disk drive components are still available separately:

BARE DRIVES — MPI drives offer the fastest track-to-track access time (5 milliseconds) available. All drives are capable of operating in double density for 80% greater storage capacity.

	Price		Freight
MPI B51 40 track, single-head, 100K	$399	New, Reduced Price	$5.00
MPI B52 40 track, dual-head, 200K	$449		$5.00
MPI B92 80 track, dual-head, 400K	$619		$5.00
Simple, wrap-around cabinet	$12		$2.00
Separate, dual-drive power supply	$85		$8.00

	Price	Freight
Self-contained, single drive cabinet/power supply	$99	$5.00
Self-contained, dual-drive cabinet/power supply	$135	$5.00
Two drive cable	$39	$2.00
Fan drive cable	$49	$2.00
DOSPLUS 3.3	$99.95	$2.00
DOSPLUS 3.4	$149.95	$2.00

Prices are FOB Adelaide. Add $5.00 freight for single drive package, $10.00 for dual-drive package. Prices are in Australian dollars. Freight is road freight anywhere in Australia.

All items carry a 90-day parts and labour warranty. Repairs to be carried out in our Adelaide workshops.

SOFTWARE BY AUSTRALIAN AUTHORS

All our software is suitable for either the SYSTEM 80 or the TRS-80

NEW SOFTWARE FROM MICRO-80 PRODUCTS

BUSINESS PROGRAMS

MICROMANAGEMENT STOCK RECORDING SYSTEM (L2/16K)

Cassette version. . . . . $29.95 + $1.00 p&p
Stringy Floppy version. . . . . $33.95 + $1.00 p&p

This system has been in use for 9 months in a number of small retail businesses in Adelaide. It is therefore thoroughly debugged and has been tailor made to suit the requirements of a small business. MICROMANAGEMENT SRC enables you to monitor the current stock level and reorder levels of 500 different stock items per tape or wafer. It includes the following features:—

- Add new items to inventory
- Delete discontinued items from inventory
- List complete file
- Search for any stock number
- Save data to cassette or wafer
- Load data from cassette or wafer
- Adjusts stock levels from sales results and receipt of goods
- List all items requiring reordering

We can thoroughly recommend this program for the small business with a L2/16K computer.

SCOTCH BRAND COMPUTING CASSETTES

Super-quality personal computing cassettes.
C-10 pack of 10 ... ... $26.00 incl. p&p
C-30 pack of 10 ... ... $28.00 incl. p&p

UTILITIES

S-KEY by Edwin Paay $15.95 plus 50c. p&p

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

S-KEY features:

* S-KEY provides an auto-repeat for all the keys on the keyboard. If any key is held down longer than about half a second, the key will repeat until it is released.
* Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80/SYSTEM 80.
* S-KEY allows text, BASIC commands and/or graphics to be defined to shifted keys. This makes programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
* Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphics symbols directly into PRINT statements.
* S-KEY allows the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
* S-KEY allows the user to list an updated list of the shift key entries to the video display or line printer.
* S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

BMON by Edwin Paay $19.95 plus 50c. p&p
THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16-48K

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays BASIC programs on the screen while they are still loading; tells you the memory locations of the program just loaded; lets you stop a load part-way through; merges two programs, with automatic renumbering of the second so as to prevent any clashes of line numbers; recovers your program even though you did type NEW: makes one program invisible while you work on a second (saves hours of cassette time!); lists all the variables used in the program; makes SYSTEM tapes; lets you Edit memory directly . . . the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone afford NOT to have BMON?

EDUCATIONAL

RPN CALCULATOR (L2/16K & 32K)
$14.95 $ 50c. p&p

Give your computer the power of a $650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for business or pleasure, this package will prove invaluable, and turn you '80 into a very powerful instrument.

GAMES

MICROPOLY (L2/16K) $8.95 + 60c p&p

Now you can play Monopoly on your micro. The old favourite board game has moved into the electronic era. This computer version displays the board on the screen, obeys all the rules and, best of all, the banker does not make mistakes with your change!

CONCENTRATION (L2/16K) $8.95 + 60c p&p

Another application of supergraphics. There are 28 "cards" displayed on the screen, face down. Players take it in turn to turn them over with the object of finding matching pairs. There are 40 different patterns which are chosen at random, so the game is full of endless variety. This is of particular value in helping young children to learn the art of concentrating and, at the same time, to introduce them to the computer.

METEOR AND TORPEDO ALLEY (L2/16K)
$10.95 + 60c p&p

Those who frequent games arcades will recognize these two electronic games. In METEOR you must destroy the enemy space ships before they see you. In its most difficult mode, the odds are a thumping 238 to 1 against you being successful. In torpedo alley you must sink the enemy ships without hitting your own supply ship. Both games include sound effects and are remarkably accurate reproductions of the arcade games.

AUSTRALIAN SOFTWARE (Cont.)

GAMES

SHEEPDOG (L2/16K) **$8.95 + 60c p&p**

Ever wondered how a sheepdog manages to drive all those awkward sheep into a pen? Well, here is your chance to find out just how difficult it is and have a lot of fun at the same time. You control the sheepdog, the computer controls the sheep! As if that isn't enough, look out for the dingoes lurking in the bush!

U BOAT **$8.95 + 60c p&p**

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedoes, plenty of targets, working fuel and battery meters, helpful Mothership for high-seas reprovisioning and even has emergency radio for that terrible moment when the depth charges put your crew at risk. Requires Level II/16K.

SPACE INVADERS WITH SOUND **$8.95 + 60c p&p**

Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

GOLF (L2/16K) **$8.95 + 60c p&p**

Pit your skills of mini-golf against the computer. Choose the level of difficulty, the number of holes and whether you want to play straight mini golf or crazy golf. Complete with hazards, water traps, bunkers and trees. Great fun for kids of all ages.

DOMINOES(L2/16K) **$8.95 + 60c p&p**

Pit your skill at dominoes against the computer, which provides a tireless opponent. Another application of supergraphics from the stable of Charlie Bartlett. Dominoes are shown approximately life size in full detail (except for colour!). The monitor screen is a window which you can move from one end of the string of dominoes to the other. Best of all, you don't lose any pieces between games!

KID'S STUFF (formerly MMM-1) **$8.95 + 60c p&p**

Three games on one cassette from that master of TRS-80 graphics, Charlie Bartlett. Includes INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNIEVEL (as in motorcycle, ramp and buses).

OTHER PROGRAMS

INFINITE BASIC BY RACET (32K/1 DISK)
$49.95 + 50c. p&p

Full matrix functions – 30 BASIC commands; 50 more STRING functions as BASIC commands.

GSF/L2/48K **$24.95 + 50c. p&p**

18 machine language routines including RACET sorts.

BUSINESS ADDRESS AND INFORMATION SYSTEM (48K/DISK) **$24.95 + 50c. p&p**

Allows you to store addresses and information about businesses, edit them and print them out.

HISPED (L216, 32 or 48K) $29.95

This machine language program allows you to SAVE and LOAD programs and data to tape at speeds up to 2000 band (4 times normal) using a standard cassette recorder. A switch must be installed to remove the XRX III loading board, if fitted.

LOWER CASE FOR YOUR TRS-80/SYSTEM 80
Kit only $49.00 plus $2.00 p&p

Give your TRS-80 or SYSTEM 80 a lower case display with proper descenders and a block cursor (similar to the TRS-80 Model III). Also includes symbols for the four suits of cards. Includes full fitting instructions, all necessary components and a special machine language driver program to enable lower case in BASIC. The modification is similar to the Tandy model and does not work with Electric Pencil without further modifications.

These kits require disassembly of your computer and some soldering. They should only be installed by someone who has experience in soldering integrated circuits, using a low power, properly earthed soldering iron. If you do not have the necessary experience/equipment, we will install the modification for you for $20 plus freight in both directions. Make sure you arrange the installation with us first, before despatching your computer, so that we can assure you of a rapid turn-around. We are also arranging to have installers in each State. See elsewhere in this issue for their names and addresses.

PRICES

Cat No.
HD-020 Lower case mod kit for TRS-80
$49.00 plus $2.00 p&p
HD-021 Lower case mod kit for SYSTEM-80
$49.00 plus $2.00 p&p

EPSON MX-80 PRINTER
ONLY *$949 Inc. Cable for TRS-80 and p&p
(*Printer only – $940 incl. p&p)

The EPSON MX-80 printer is compact, quiet, has features unheard of only 2-3 years ago in a printer at any price and, above all, is ultra-reliable. All available print modes may be selected under software control. Features include:

- high quality 9x9 dot-matrix character formation
- 3 character densities
  - 80 characters per line at 10 chars/inch
  - 132 characters per line at 16.5 chars/inch
  - 40 characters per line at 5 chars/inch
- 2 line spacings
  - 6 lines per inch 8 lines per inch
- 80 characters per second print speed
- bi-directional printing
- logical seeking of shortest path for printing
- lower case with descenders
- TRS-80 graphics characters built in
- standard Centronics printer port

The bi-directional printing coupled with the logical seeking of the shortest print path (which means that the print head will commence printing the next line from the end which requires the least travel, thereby minimising unutilised time) gives this printer a much higher throughput rate than many other printers quoting print speeds of 120 c.p.s. or even higher.

GREEN SCREEN SIMULATOR
$9.50 incl. p&p

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image.

All editorial staff of MICRO-80 are now using GREEN SCREEN SIMULATORS on their own monitors.

Please make sure to specify whether you have an old (squarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

Line 1620 checks for a right subtree and traverses this in the same manner as for the Left subtree.

Line 1630 checks to see if the process is complete (stack empty). If not, line 1640 pops the top of the stack, being the Root node of a subtree to be processed in the above manner.

Using the data in the structure in Figure 1 (B), the items could be written in linear fashion under the inorder traversal principle as follows, using nested parentheses to show the Root/Subtrees structure.

((23) 26 ((33) 38)) 42 (53 ((58) 69 (69)))

From this it can be seen that 42 (main root) has no parenthesis (level zero) and has two groups of items to its left and right (subtrees). In fact, the level of parenthesis nesting for any given item in that list corresponds to its level in the tree in Figure 1 (B), where level zero is the main root. Well, lo and behold! The list as output using the INORDER traversal principle gives us an ascending sorted list of the items that were built into the tree.

THE COMPLETE TREE SORT

The complete sort with driver, tree building and tree traversal modules follows:

```
2110 CLS:DEFINTA-Z:INPUT"SIZE OF LIST TO BE SORTED";LM
2120 IFLM<1THEN2110
2130 TRUE=-1:FALSE=0
2150 DIMT(LM,3),OL(LM),S(LM/2)
2160 PRINT"LIST BEFORE SORT:"
2170 FORM=1TOLM
2180 T(M,2)=RND(1000)
2190 PRINTT(M,2);
2200 NEXT:PRINT:INPUT"HIT ENTER TO START THE SORT";X
2210 PRINT" SORTING...":GOSUB2260
2220 PRINT"LIST AFTER SORT: "
2230 FORM=1TOLM:PRINTOL(M);:NEXT
2240 END
2250 '    CONTROLLING ROUTINE
2260 IFLM=0PRINT" INVALID LIST SIZE":GOTO2290
2270 IFLM=1OL(1)=T(1,2):GOTO2290
2280 GOSUB2310:GOSUB2390
2290 RETURN
2300 '    CREATION OF BINARY TREE
2310 FORI=2TOLM
2320 LINKED=FALSE:P=1
2330 IF LINKED THEN 2360
2340 IFT(I,2)<T(P,2)THEN IFT(P,1)=0THEN T(P,1)=I:LINKED=TRUE ELSE P=T(P,1) ELSE
IFT(P,3)=0THEN T(P,3)=I:LINKED=TRUE ELSE P=T(P,3)
2350 GOTO2330
2360 NEXT
2370 RETURN
2380 '   INORDER TRAVERSAL OF BINARY TREE
2390 P=1:SP=0:OP=0
2400 IFT(P,1)<>0THENSP=SP+1:S(SP)=P:P=T(P,1):GOTO2400
2410 OP=OP+1:OL(OP)=T(P,2)
2420 IFT(P,3)<>0THENP=T(P,3):GOTO2400
2430 IFSP=0GOTO2460
2440 P=S(SP):SP=SP-1
2450 GOTO2410
2460 RETURN
```

EFFICIENCY

The efficiency of the tree sort algorithm is of the order of N* (log N to BASE 2), similar to the diminishing increment sort analyzed last month. However, this varies with the actual values of the data. In cases where the data is such that a fairly balanced tree is built, then it would be fairly efficient, maybe even more so than the diminishing increment sort. However, if the tree is severely unbalanced (i.e. a greater proportion of Left subtrees than Right subtrees, or vice versa, known as left or right-heavy), then the tree structure approaches that of a linked linear list, since only one of the pointers in a node is being made use of most of the time. When this is the case, then the efficiency degrades to that similar to the selection sort. Such a case could occur if the data is already mainly presorted (ascending or descending). If the data is fairly random, then the sorting efficiency will be quite high. Other tree sorting algorithms exist which have different efficiency properties, one other known as the "heapsort". However, none could be described as being of markedly greater efficiency, as their characteristics depend largely on the data itself.

TO SUMMARIZE

A fairly efficient sorting technique can be devised using a tree data structure. One such algorithm builds a network of relationships between the data, without actual data movement itself, to form a tree structure of the data. Then the tree is traversed using the INORDER principle, to obtain the final sorted output as the nodes are being processed. A stack is required in the traversal process unless the traversal is being performed recursively.

Next month, a sorting algorithm based on sorting "by exchange" that really is "quick" will be presented.

- 0000000000 -

* * * * SOFTWARE SECTION * * * *

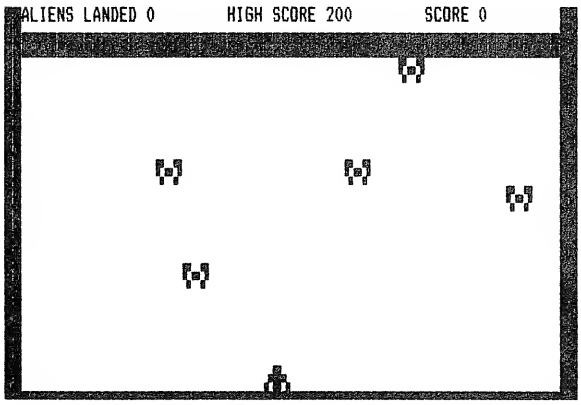
***** ALIEN INVASION LII/16K - by J. Triffitt *****

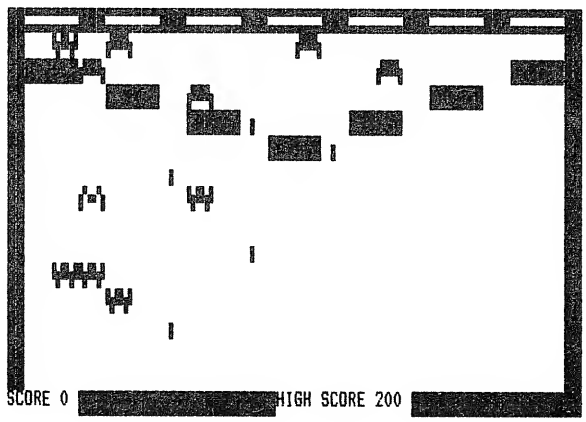
This is a real time graphics game written in Level Two BASIC and consists of two parts.

In the first part, the aliens space craft attack from above. In the second part, the aliens fight after five of their ships have landed.

For fast graphics POKE is used in preference to SET. To move the enemy there is a FOR-NEXT loop used with the array function. This is sped up by replacing it with a variable if it is used frequently, e.g. instead of -

```
                                FORI=1TO10:POKEG(I),191:NEXTI
(The alternative)       FORI=1TO10:A=G(I):POKEA,191:NEXTI       is used.
```





The enemy moves a random distance in a random direction before changing course, in the first part of the game. To use keyboard controls the INKEY$ command is used. The wall around the screen in both parts is used to avoid objects moving into a different part of the memory, since objects will see the wall with the aid of PEEK and rebound off it. PEEK is also used to check for the collision of objects with other objects. The direction in which an object moves is determined by a variable which is added onto the value of the object's position each time it is moved. To save the time waste of continually checking for every object on one particular byte to see if something is there, the program checks first to see if it is an empty byte and if not, then what is there since the majority of bytes will be empty. The tracking aliens in the second part know if you are above or below them by using the equation where X is your position and Y is theirs

```
A=INT(X/64)
B=INT(Y/64)
```

and then finding out whether A or B is the greater number. They know if you are to the right or left of them by the equations, where X is your position and Y is theirs

```
A=INT(X/64)*64
B=INT(Y/64)*64
```

and then finding out whether A or B is the greater number.

- 0000000000 -

***** WHEEL LOADER PRODUCTION LI/4K - by G.T. Lyon *****

This program is designed to calculate Wheel Loader Production in tons per hour (1 ton = 2000 lb.). It can be used for all sizes of loaders in all types of conditions.

Input data is requested to cover all variables and the result is calculated in tons per hour. The production which is based on average truck loading conditions in average digging material, can be used by contractors for estimating job costs and for salesmen recommending average machine capabilities.

The time to complete the job can be worked out and loader size and number can be used to establish completion time for the job with a given number and size of machine.

Line 40 Requests the in bank weight per cu yd of the material.
i.e. Earth, Damp Loose = 2106 lbs.
Gravel, Dry = 2970 lbs.
Coal Bituminous = 1350 lbs.

Line 50 Requests in bank correction factor
i.e. Earth, Damp Loose = .85
Dry Gravel = .89
Coal Bituminous = .74

Line 60 Requests the heaped capacity (S.A.E. Rated capacity of the loader.
i.e. This is mid way between the struck capacity and the heaped.

Line 70 Requests the efficiency hour.
i.e. Usually use a 50 MT Hour. Other factors may reduce this to a 40 or 45 MT Hour.

Line 80 Requests Cycle time.
i.e. Favourable conditions = .30 MT
Average conditions = .33 to .50 MT
Unfavourable conditions = 0.42 to 0.66 MT

Line 90 This is the formula, based on proving ground tests and field experience of many years.

- 0000000000 -

***** RESTORE (LINE NUMBER) LII/16K - by W. Eldridge *****

INTRODUCTION

Imagine, if you can, the problems of writing BASIC programs if you were denied the use of GOTO or GOSUB statements or, machine language with no JUMP or CALL facility. Impossible? Well, similar restrictions are placed on us when it comes to reading DATA because we only have the single RESTORE function available to us. This means that to read a particular piece of DATA we have to restore the DATA POINTER to the start of the BASIC program and then read all intervening DATA between the start of the program and the DATA we want. On some personal computers, the HITACHI PEACH to name one, it is possible to include in the program a statement such as RESTORE 300. This then sets the DATA POINTER to look at line 300 when the program encounters a READ statement. I am pleased to announce that this facility is now available on the TRS-80 by including a 13 byte machine language program in your BASIC and using X=USR(LN).

EXPLANATION

1. The DATA POINTER is in reserved RAM at 40FF and 4100.
2. Every BASIC line starts and ends with 00, and this can be used as an originator, a separator or a terminator.
3. ROM routine 0A7F takes the value contained in brackets in the statement X=USR(0), converts it to an integer and returns with the value in the HL register pair.
4. ROM routine 1B2C searches through a BASIC program to find a line number matching the value in the DE register pair. When a match is found, the BC register pair contains the address of the start of the line.
5. The DATA POINTER always points to an address that contains either 00H (the beginning or end-of-line marker) or 2CH (which is the ASCII value for a comma used as a separator in DATA lines).
6. The DATA POINTER normally points to 42E8H (which contains 00H) immediately after a RESTORE has been executed, or the computer is in the COMMAND mode.

If we type NEW and insert the following line,

10 DATA12, 14, 16:READ A:READ B:STOP (ENTER)

then use a monitor, such as BMON, to examine memory we see the following:

```
40FF    (E8)      :LSB of DATA POINTER
4100    (42)      :MSB "    "    "
42E8    (00)      :Line originator.
42E9    (01)      :LSB of start address of the next line.
42EA    (43)      :MSB "    "    etc.
```

```
42EB    (0A)      :LSB of this line number in Hex.
42EC    (00)      :MSB "    "   etc.
42ED    (88)      :Hex compression code for DATA.
42EE    (31)      :ASCII code for 1 expressed in Hex.
42EF    (32)      :ASCII code for 2 expressed in Hex.
42F0    (2C)      :ASCII code for a (,).
42F1    (31)      :ASCII 1.
42F2    (34)      :  "   4.
42F3    (2C)      :Comma.
42F4    (31)      :ASCII 1.
42F5    (36)      :ASCII 6.
42F6    (3A)      :Colon (:). Statement separator.
42F7    (8B)      :Hex compression code for READ.
42F8    (20)      :Space.
42F9    (41)      :ASCII code for A.
42FA    (3A)      :Colon (:).
42FB    (8B)      :READ.
42FC    (20)      :Space.
42FD    (42)      :ASCII B.
42FE    (3A)      :Colon (:).
42FF    (94)      :ASCII code for STOP.
4300    (00)      :Line separator.
4301    (00)      :This is where the next line would begin if
there was one. See 42EA and 42E9 at the start of this listing.
```

As all BASIC lines follow the above format, it is probably obvious by now that if we can find where in memory the line that contains the DATA we need is located we can load the DATA POINTER to start at this line when the program encounters a READ statement. This is the machine language program that achieves it.

```
CD7F0A      CALL 0A7F       : Take the variable LN from X=USR(LN), convert it to an integer
                              and return with it in the HL register pair.
EB          EX DE,HL        : Place LN in the DE register pair.
CD2C1B      CALL 1B2C       : Search BASIC program for a line number matching LN and return
                              with the start address in the BC register pair.
0B          DEC BC          : Decrement BC to point to 00H at the beginning of line number
                              LN.
ED43FF40    LD(40FF),BC     : Set DATA POINTER to read LN on encountering the next READ statement.
C9          RET.            : Return to BASIC program.
```

IMPLEMENTATION

The accompanying BASIC program was written to test and demonstrate the effectiveness of the above program. The information contained in the DATA lines is taken from the TELECOM AUSTRALIA publication, "S T D Area Codes and Information For the Adelaide Telephone District 1979". The information in this program contains every place name in SA/NT and Tasmania plus all place names in N.S.W. commencing with the letter B. The letter B was chosen because it comprises the largest single alphabetical entry in the book, (186 names) and, as this program is designed to search only one alphabetical block at a time, it can be demonstrated that a place name, together with its STD Area code and cost code letter can effectively be retrieved from 36 pages of entries in under 5 seconds. To further test the program, the N.S.W. B entries have been sandwiched between the Q and R entries for SA/NT.

The program functions as follows:

Lines 12 and 13 contain a small program to take the DATA contained in line 11 and transfer it to the dummy REMARK line, line 10.

Line 14 then deletes lines 11 through 14 so that initialisation occurs only once. Note: At this point the program can be CSAVED and line 10 will contain the machine language program.

Lines 30 through 80 contain the line numbers where the 26 letters of the 5 States are located. This DATA is used in line 90 to initialise the 2 dimensional array AR(R,C).

Line 110 takes the ASCII value of the first letter of the place name and subtracts 65, to give a value for R of 0-25 representing the letters A-Z. It also takes the ASCII value of the first letter of the State and produces the following values for C. QLD=0..N.S.W.=1..SA/NT=2..TAS=3.. VIC=4.. W.A.=5.

Line 120 uses these values to get a value for LN from the array (AR(R,C)).

Line 130 passes this value to the machine language program via the POKE and X=USR(LN). The DATA POINTER is then set as previously explained.

Line 140 and 150 search the DATA commencing at the nominated line until a match is found for the place name or the terminator ("9999") is encountered. If a match is found the details are printed to the screen otherwise "NOT LISTED.." is printed.

Line 160 waits for a key to be pressed then prompts for another entry.

NOTE!!!!! Performing a Re-number will render the whole program inoperative unless the data in lines 30 through 80 is changed.

CONCLUSION

While this program may be rather simple, I feel that it opens up some interesting possibilities for the rapid access to large quantities of DATA. Several enhancements to the accompanying BASIC program come to mind which may further increase the speed of operation. One of these may be, on initialisation, to systematically go through the array and have each line number searched for and replaced, in the array, by the memory address of that line. This would mean then that the DATA POINTER could have that address POKEd into it via BASIC, (possibly not very fast) or another small machine language program could be used with X=USR(AD) or similar. A suitable program could be

```
CD7F0A          CALL 0A7F
22FF40          LD(40FF),HL
C9              RET.
```

- 0000000000 -

***** SOLITAIRE PROGRAM PATCH LII/16K - by M. Warrington *****

In regard to SOLITAIRE by T. Griffin, September 81 (issue 22). Imagine you have been playing this game. Suppose ! just suppose you get it right. Could you do it again? With this minor addition to the program you will be able to review all your moves and, hopefully, learn enough to be able to win every time.

- 0000000000 -

***** LEMNISCATES LII/4K - by Ken Shillito *****

This program is based on a gimmick, viz., how much could be done by a program just one line long? What it does is to plot curves from the family of curves related to the lemniscate, given any two numbers as parameters. You can experiment with any numbers and by all means, try ones like, say, 1E6 or 1E-6 as well as more commonplace ones.

You will be surprised by the variety of effects. If you input 1,2 you get a true lemniscate. A lemniscate is a sort of muddled up ellipse, and the curve was used by its inventor, Bernouilli in the 18th century in a vain effort to measure the length of an ellipse. We have now progressed to the stage of proving that impossible.

Here are some interesting number pairs to try for starters:

1,1 33,1 100,1 2,3 2,5 5,10 100,100 667,667 250,250
100,.1 1E-6,1E-6 5,7

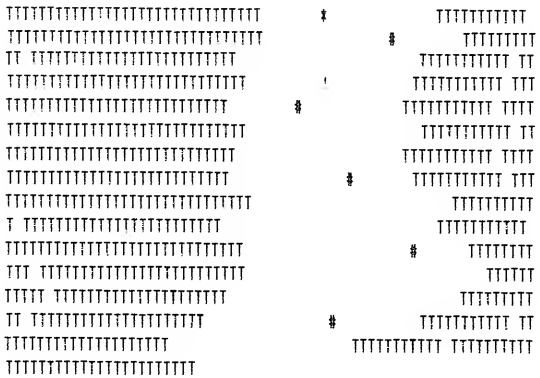
And remember, that's just a sample.

- 000000000 -

***** PUNCTUATION LII/16K - by Ken Shillito *****

Here is a challenging game that requires quick reflexes and an ability to think ahead. Can you get yourself promoted to the rank of circumflex? The best that I have managed so far is a > a few times. My 10 year old son beats me at it! The exclamation marks have a limited ability to penetrate the walls, so if ever you get "stuck" you can gauge your way through if you use the / key carefully.

You are an asterisk which can move from side to side. Use the < and > keys to move sideways. Occasionally, fearsome # signs come towards you. If they hit you, or you hit the column wall, you die horribly. You can zap #'s with !'s, which kill #'s on contact. To fire an ! press / - but only one ! can exist at a time. The page gets more tortuous and the #'s progressively denser. Eventually, you will die. You will get a posthumous score for the #'s killed. If you score well enough, you will re-incarnate as a < or even a > - and if you're a real whiz, as a circumflex !!!!! Sic transit gloria mundi.



VIEW OF "PUNCTUATION" SCREEN

*****ALIEN INVASION L2/16K*****

```
10 REM              JONATHAN TRIFFITT
20 REM              20 LOFTUS ST. NEDLANDS PERTH W.A.   6009
                    PH. (09) 3866964
30 XU=5:CLS:PRINT@467,"ALIEN INVASION":FORR=1TO5:FORW=1TO3:IFW=1
THENA$="+":GOTO50
40 IFW=2THENA$="X"ELSEA$="*"
50 FORZ=1TO14:PRINT@530+Z,A$:NEXTZ:NEXTW:NEXTR:A$=""
60 PRINT@640,"DO YOU WANT INSTRUCTIONS":INPUTZ$:IFLEFT$(Z$,1)="Y
"THEN2300ELSE70
70 CLS:FOR R=16320TO16383:POKER,131:NEXTR:FORR=15424TO15485:POKE
R,191:NEXTR:FORR=15360TO16256STEP64:POKER,191:POKER+63,191:POKER+
1,191:POKER+62,191:NEXTR:R=0
80 F=0:Y=0:Z=0:A=16286:PRINT@2,"ALIENS LANDED";F;:PRINT@47,"SCOR
E";:F(1)=12:F(2)=21:F(3)=30:F(4)=42:F(5)=51:PRINT@25,"HIGH SCORE"
;HS;
90 PRINT@52,S;:IFXX=1THENXX=0:GOTO790ELSED4=D4+1:OND4GOSUB720,73
0,740,750,760:FORI=1TO5:V(I)=0:G(I)=0:B(I)=1:A(I)=15488+F(I)+(D4-
1)*128:G9=1:GOSUB600:GOSUB810:NEXTI:POKEA,191:POKEA+1,164:POKEA-1
,152
100 Z=0
110 GOTO480
120 IF A$="," THENY=2:GOTO210
130 IFA$="."THENY=1:GOTO180
140 IFA$="M"ANDZ=0THEN280
150 IFA$="/"THENY=0:GOTO640
160 ONYGOTO180,210
170 GOTO640
180 IFPEEK(A+2)>128THEN260
190 A=A+3:POKEA,191:POKEA-4,128:POKEA-3,128:POKEA-1,152:POKEA+1,
164:POKEA-2,128
200 GOTO 640
210 IFPEEK(A-2)>128THEN240
220 A=A-3:POKEA,191:POKEA+4,128:POKEA+3,128:POKEA-1,152:POKEA+1,
164:POKEA+2,128
230 GOTO 640
240 IFPEEK(A-2)=191THENY=0:GOTO480
250 A=A-3:POKEA,191:POKEA+4,128:POKEA+3,128:POKE-1,152:POKEA+1,1
64:POKEA+2,128:GOTO440
260 IFPEEK(A+2)=191THENY=0:GOTO480
270 A=A+3:POKEA,191:POKEA-4,128:POKEA-3,128:POKEA-1,152:POKEA+1,
164:POKEA-2,128:GOTO440
280 Z=5:C1=A:IFPEEK(C1-64)=Z5THEN670ELSEC1=C1-64:POKEC1,148
290 FORV7=1TO3:IFPEEK(C1-64)>128THEN340
300 C1=C1-64:POKEC1,148
310 POKEC1+64,128
320 NEXTV7
330 GOTO650
340 IFPEEK(C1-64)=Z5THEN670ELSEPOKEC1,128:C1=0:Z=0:GOTO480
350 L2(I)=0:L3(I)=0:IFPEEK(X5)=148THENPOKEX5,Z5:POKEX5-1,Z4:POKE
X5+1,Z6:A(I)=X5:POKEA(I)-G(I),128:POKEA(I)-G(I)-1,128:POKEA(I)-G(
I)+1,128:GOTO680
360 X5=A(I)+G(I)
370 IFX5=ATHEN420
380 IFPEEK(X5)=131THEN770
390 L2(I)=1:IFABS(G(I))=61THENG(I)=67:IFI<5THEN650ELSE480
400 IFABS(G(I))=67THENG(I)=61ELSEG(I)=0-G(I)
410 IFI<5THEN650ELSE480
420 POKEA(I)+G(I),Z5:POKEA(I)+G(I)+1,Z6:POKEA(I)+G(I)-1,Z4
430 POKEA(I),128:POKEA(I)-1,128:POKEA(I)+1,128
440 POKEA,140:FORU=1TO20:NEXTU:POKEA,146:POKEA-1,136:POKEA+1,132
:FORU=1TO20:NEXTU:POKEA,128:POKEA+1,162:POKEA-1,145:FORU=1TO20:NE
XTU
450 POKEA,128:POKEA-1,128:POKEA+1,128
460 IFC2=1THENC2=0:A(I)=0:A=C:GOTO690
470 M=960:GOTO2270
480 FORI=1TO5:IFL2(I)=1THEN580
490 IFB(I)=0THEN650
500 L(I)=RND(2):L2(I)=1:L(I)=L(I)*5
510 R=RND(8):ONRGOTO520,520,530,530,540,550,560,570
520 G(I)=67:GOTO580
530 G(I)=61:GOTO580
540 G(I)=-61:GOTO580
550 G(I)=-67:GOTO580
560 G(I)=-3:GOTO580
570 G(I)=3
580 X5=A(I)+G(I):L3(I)=L3(I)+1:IFL3(I)=L(I)THENL3(I)=0:L2(I)=0
590 IFPEEK(X5)>128THEN350
600 A(I)=A(I)+G(I):X6=A(I):POKEX6,Z5:POKEX6+1,Z6:POKEX6-1,Z4:IFG
9=1THENG9=0:RETURN
610 POKEX6-G(I),128:POKEX6-G(I)+1,128:POKEX6-G(I)-1,128
620 A$=INKEY$:IFA$<>""THEN120
630 ONYGOTO180,210
```

```
640 IFZ=5THEN290
650 NEXTI
660 GOTO480
670 FORI=1TO5:IFC1-64=A(I)THENL(I)=0:L3(I)=0:L2(I)=0ELSE700
680 S=S+S2:B(I)=0:C=A:A=A(I):C2=1:IFC1<CTHENPOKEC1,128:GOTO430EL
SE430
690 C1=0:Z=0:PRINT@52,S;:IFB(1)=0ANDB(2)=0ANDB(3)=0ANDB(4)=0ANDB
(5)=0THEN90ELSE480
700 NEXTI
710 RETURN
720 Z4=167:Z5=140:Z6=155:S2=50:RETURN
730 Z4=166:Z5=140:Z6=153:S2=100:RETURN
740 Z4=151:Z5=140:Z6=171:S2=150:RETURN
750 Z4=157:Z5=179:Z6=174:S2=200:RETURN
760 Z4=167:Z5=140:Z6=155:S2=50:D4=1:RETURN
770 POKEA(I),128:POKEA(I)+1,128:POKEA(I)-1,128:F=F+1:PRINT@15,F;
:G(I)=0:XX=1:V(I)=1:B(I)=0:IFF=>5THEN860
780 IFB(1)=0ANDB(2)=0ANDB(3)=0ANDB(4)=0ANDB(5)=0THEN90ELSE620
790 Y=0:Z=0:POKEC1,128:FOR I=1TO5:IFV(I)=1THENV(I)=0:B(I)=1:A(I)
=15488+F(I)+(D4-1)*128:POKEA(I),Z5:POKEA(I)-1,Z4:POKEA(I)+1,Z6:GO
SUB 810
800 NEXTI:GOTO480
810 L2(I)=1:L(I)=5:L3(I)=0:R=RND(4):ONRGOTO820,830,840,850
820 G(I)=3:RETURN
830 G(I)=-3:RETURN
840 G(I)=67:RETURN
850 G(I)=61:RETURN
860 FORW=1TO500:NEXTW:CLS:PRINT"THE SPACECRAFT HAVE LANDED.YOU M
UST NOW FIGHT THE INVADING      ALIENS TO SAVE EARTH!":FORT=16383
TO15360STEP-1:POKET,128:NEXTT
870 A=0:L=0:H=0:B=0:I=0:R=0:Q=0:T=0:M=0:W=0:Z=0:Y=0:X=0:G=0:P=0:
J=54:C=5:FORN=1TO4:FORA=15490+H+LTO15544-H+LSTEPJ:FORB=ATOA+C:POK
EB,191:NEXTB:NEXTA:L=L+64:H=H+9:J=J-18:NEXTN
880 A(1)=15427:A(2)=15445:A(3)=15466:A(4)=15481:A(5)=15500:A(6)=
15536:A(7)=15573:A(8)=15594:A(9)=15643:A(10)=15649
890 FORJ=15360TO15423:POKEJ,179:POKEJ+960,191:NEXTJ:FORJ=15360TO
16256STEP64:POKEJ,191:POKEJ+1,191:POKEJ+62,191:POKEJ+63,191:NEXTJ
:J=0:MO=0
900 FORI=1TO10:POKEA(I),143:POKEA(I)+1,173:POKEA(I)-1,158:H(I)=0
:K(I)=0:EB(I)=0:Q(I)=0:B(I)=0:NEXTI:PRINT @960,"SCORE";S;:PRINT@9
90,"HIGH SCORE";HS;
910 XU=XU-1:IFXU=0THENXU=1:FORI=1TO10:B(I)=2:NEXTI
920 N=0:Q=9:X=16161:T=-64:FORZ8=1TO10:U(Z8)=3:NEXTZ8:FORR=15367T
O15423STEPQ:FORL=1TO3:POKER+L,191:NEXTL
930 NEXTR:Q=0:L=0:R=0:N=0:TN=2:TM=0:M=0:MO=0
940 Z(1)=64:Z(2)=64:Z(3)=64:Z(4)=64:Z(5)=128:Z(6)=128:Z(7)=192:Z
(8)=192:Z(9)=256:Z(10)=256
950 POKEX-1,150:POKEX,140:POKEX+1,169
960 GOTO1380
970 IFA$=".":THENMO=1:RETURN
980 IFA$=",":THENMO=2:RETURN
990 IFA$=";":THENMO=4:RETURN
1000 IFA$="@":THENMO=3:RETURN
1010 IFA$="/":THENMO=0:RETURN
1020 IFA$="M"ANDJ=0THENJ=1:GOTO1320
1030 RETURN
1040 T=3
1050 IFPEEK(X+3)<>128THEN1080
1060 X=X+3:POKEX-4,128:POKEX-3,128:POKEX-2,128:POKEX-1,150:POKEX
,140:POKEX+1,169
1070 RETURN
1080 IFPEEK(X+3)=148ORPEEK(X+3)=143THEN1100
1090 MO=0:RETURN
1100 X=X+3:POKEX-4,128:POKEX-3,128:POKEX-2,128:POKEX-1,150:POKEX
,140:POKEX+1,169:GOTO2070
1110 T=-3
1120 IFPEEK(X-3)<>128THEN1150
1130 X=X-3:POKEX+4,128:POKEX+3,128:POKEX+2,128:POKEX-1,150:POKEX
,140:POKEX+1,169
1140 RETURN
1150 IFPEEK(X-3)=148ORPEEK(X-3)=143THEN1170
1160 MO=0:RETURN
1170 X=X-3:POKEX+4,128:POKEX+3,128:POKEX+2,128:POKEX+1,169:POKEX
,140:POKEX-1,150:GOTO2070
1180 T=-64
1190 IFPEEK(X-64)<>128THEN1220
1200 X=X-64:POKEX+63,128:POKEX+64,128:POKEX+65,128:POKEX-1,150:P
OKEX,140:POKEX+1,169
1210 RETURN
1220 IFPEEK(X-64)=143ORPEEK(X-64)=148THEN1240
1230 MO=0:RETURN
1240 X=X-64:POKEX+63,128:POKEX+64,128:POKEX+65,128:POKEX-1,150:P
OKEX,140:POKEX+1,169:GOTO2070
1250 T=64
1260 IFPEEK(X+64)<>128THEN1290
1270 X=X+64:POKEX-63,128:POKEX-64,128:POKEX-65,128:POKEX-1,150:P
OKEX,140:POKEX+1,169
1280 RETURN
1290 IFPEEK(X+64)=143ORPEEK(X+64)=148THEN1310
1300 MO=0:RETURN
1310 X=X+64:POKEX-63,128:POKEX-64,128:POKEX-65,128:POKEX-1,140:P
OKEX,150:POKEX+1,169:GOTO2070
1320 G=X+T:O=T
1330 IFPEEK(G)<>128THEN1580ELSEPOKEG,42
1340 G=G+O
1350 IFPEEK(G)<>128THEN1570
1360 POKEG,42:POKEG-O,128
1370 RETURN
1380 FORI=1TO10
1390 M=A(I):IFH(I)=1THENF=K(I)+EB(I):GOSUB1760
1400 ONB(I)GOTO1480,1970
1410 IFH(I)=0ANDPEEK(M-Z(I))=191ANDPEEK(M+64)=128THENK(I)=M+64:P
OKEK(I),148:F=M+128:EB(I)=64:H(I)=1:GOSUB1760
1420 N=A(I)+U(I):IFPEEK(N)<>128THEN1500
1430 POKEM-1,128:POKEM,128:POKEM+1,128:A(I)=N:POKEN,143:POKEN+1,
173:POKEN-1,158
1440 A$=INKEY$:IFA$<>""THENGOSUB970
1450 IFJ=1THENGOSUB1340
```

```
1460 A$=INKEY$:IFA$<>""THENGOSUB970
1470 TM=TM+1:IFTM=TNTHENTM=0:ONMOGOSUB1040,1110,1180,1250
1480 NEXTI
1490 GOTO1380
1500 Q(I)=Q(I)+1:IFQ(I)=XUTHENB(I)=2
1510 IFPEEK(N)<>143THEN1540
1520 IFU(I)=-3THENU(I)=3:U(I-1)=-3ELSEU(I)=-3:U(I+1)=3
1530 GOTO1480
1540 IFPEEK(N)=140THEN2070
1550 U(I)=0-U(I)
1560 GOTO1480
1570 POKEG-0,128
1580 IFPEEK(G)=143THEN1590ELSEG=0:J=0:RETURN
1590 FORY=1TO10
1600 IFG=A(Y)THEN1620
1610 NEXTY
1620 E=A(Y):A(Y)=0:POKEG,128:G=0:L2(Y)=0:B(Y)=1:S=S+50:POKEE,128
:POKEE+1,128:POKEE-1,128
1630 FORR=1TO4:POKEE-1,144:POKEE-1,160:POKEE-1,128:POKEE,144:POK
EE,160:POKEE+1,144:POKEE,128:POKEE+1,160:POKEE+1,136:POKEE+1,130:
POKEE+1,129:POKEE,130:POKEE+1,128:POKEE,129:POKEE,128:POKEE-1,130
:POKEE-1,129:POKEE-1,132:POKEE-1,136:POKEE,132:POKEE-1,128
1640 POKEE,136:POKEE,128
1650 POKEE,132:POKEE-1,136:POKEE,128:POKEE-1,132:POKEE-1,129:POK
EE-1,130:POKEE,129:POKEE-1,128:POKEE,130:POKEE+1,129:POKEE,128:PO
KEE+1,130:POKEE+1,136:POKEE+1,160:POKEE+1,144:POKEE,160:POKEE+1,1
28:POKEE,144:POKEE-1,160:POKEE,128:POKEE-1,140
1660 NEXTR
1670 POKEE,128:POKEE+1,128:POKEE-1,128
1680 IFH(Y)=1THENPOKEK(Y),128
1690 H(Y)=1:K(Y)=E:POKEE,148:EB(Y)=0-0
1700 FORDT=1TO10:IFB(DT)=1THENDU=DU+1
1710 NEXTDT
1720 IFDU=10THENDU=0:XX=0:C6=0:GOTO70
1730 IFDU=8THENDU=0:TN=1:TM=0
1740 DU=0
1750 J=0:PRINT@965,S;:RETURN
1760 IFPEEK(F)=128THEN1800
1770 IFPEEK(F)=42THENJ=0:G=0:GOTO1800
1780 IFPEEK(F)<>140THENPOKEF-EB(I),128:H(I)=0:K(I)=0:RETURN
1790 K(I)=F:POKEF,148:POKEF-EB(I),128:GOTO2070
1800 POKEF,148
1810 POKEF-EB(I),128:K(I)=F:RETURN
1820 IFTZ<0THEN1900
1830 IFTZ<64THEN1870
1840 IFPEEK(M-3)=128THENM=M-3:A(I)=M:POKEM,143:POKEM+1,158:POKEM
-1,173:POKEM+4,128:POKEM+2,128:POKEM+3,128:GOTO1440
1850 IFPEEK(M-64)=128THENM=M-64:A(I)=M:POKEM,143:POKEM+1,158:POK
EM-1,173:POKEM+63,128:POKEM+64,128:POKEM+65,128:GOTO1440
1860 GOTO1440
1870 IFPEEK(M+3)=128THENM=M+3:A(I)=M:POKEM,143:POKEM+1,158:POKEM
-1,173:POKEM-4,128:POKEM-3,128:POKEM-2,128:GOTO1440
1880 IFPEEK(M-64)=128THENM=M-64:A(I)=M:POKEM,143:POKEM+1,158:POK
EM-1,173:POKEM+63,128:POKEM+64,128:POKEM+65,128:GOTO1440
1890 GOTO1440
1900 IFTZ>-64THEN1940
1910 IFPEEK(M+3)=128THENM=M+3:A(I)=M:POKEM-1,173:POKEM,143:POKEM
+1,158:POKEM-3,128:POKEM-4,128:POKEM-2,128:GOTO1440
1920 IFPEEK(M+64)=128THENM=M+64:A(I)=M:POKEM,143:POKEM+1,158:POK
EM-1,173:POKEM-65,128:POKEM-64,128:POKEM-63,128:GOTO1440
1930 GOTO1440
1940 IFPEEK(M-3)=128THENM=M-3:A(I)=M:POKEM,143:POKEM+1,158:POKEM
-1,173:POKEM+4,128:POKEM+2,128:POKEM+3,128:GOTO1440
1950 IFPEEK(M+64)=128THENM=M+64:A(I)=M:POKEM,143:POKEM+1,158:POK
EM-1,173:POKEM-64,128:POKEM-65,128:POKEM-63,128:GOTO1440
1960 GOTO1440
1970 IFINT(M/64)<INT(X/64)THENM=M+64:A(I)=M:TZ=-64ELSETZ=64:M=M-
64:A(I)=M
1980 IFM-INT(M/64)*64<X-INT(X/64)*64THENM=M+3:A(I)=M:TZ=TZ-3ELSE
M=M-3:A(I)=M:TZ=TZ+3
1990 IFPEEK(M)=128THEN2050
2000 IFPEEK(M)=143THENA(I)=M+TZ:GOTO1440
2010 IFPEEK(M)=148THENA(I)=M+TZ:GOTO1440
2020 IFPEEK(M)=42THENY=I:POKEM,143:POKEM+1,158:POKEM-1,173:POKEM
+TZ-1,128:POKEM+TZ,128:POKEM+TZ+1,128:GOSUB1620:GOTO1440
2030 IFPEEK(M)<>140THENM=M+TZ:A(I)=M:GOTO1820
2040 POKEM,143:POKEM+1,158:POKEM-1,173:POKEM+TZ-1,128:POKEM+TZ,1
28:POKEM+TZ+1,128:GOTO2070
2050 POKEM,143:POKEM+1,158:POKEM-1,173:POKEM+TZ-1,128:POKEM+TZ,1
28:POKEM+TZ+1,128
2060 GOTO1440
2070 POKEX,128:POKEX-1,128:POKEX+1,128:POKEX+63,128:POKEX+64,128
:POKEX+65,128:M=0
2080 FORR=1TO5:POKEX,128:POKEX+1,128:POKEX-1,128:POKEX,140:FORU=
1TO10:NEXTU:POKEX,146:POKEX-1,136:POKEX+1,132:FORU=1TO10:NEXTU:PO
KEX,128:POKEX+1,162:POKEX-1,145:NEXTR
2090 POKEX,160:POKEX+1,132:POKEX,128:POKEX+1,130:A=X+1:POKEA,128
2100 POKEX,188:POKEX-1,166:POKEX+1,157:
2110 IFPEEK(A-63)<>191ANDPEEK(A-63)<>179THENREADBELSEPOKEA,128:G
OTO2140
2120 A=A-63:POKEA,B:POKEA+63,128:IFB=130THENRESTORE:GOTO2110ELSE
2110
2130 DATA 144,136,129,160,132,130
2140 POKEX+63,149:IFPEEK(X+126)=191ORPEEK(X+126)<128THEN2190
2150 POKEX-1,128:C=X-2:FORE=1TO10:NEXTE:POKEC,170:IFPEEK(C-1)=19
1THEN2190ELSEFORE=1TO10:NEXTE:POKEC-1,137:POKEC,144
2160 IFPEEK(C-2)=191THEN2190ELSEFORE=1TO10:NEXTE:POKEC-2,176:POK
EC-1,176:POKEC,176
2170 IFPEEK(C+63)=191ORPEEK(C+63)<128THEN2190ELSEFORE=1TO10:NEXT
E:POKEC,144:POKEC-1,128:POKEC-2,128:POKEC+63,134:C=C+64:FORE=1TO1
0:NEXTE:POKEC,170:POKEC-1,128:POKEC-64,128
2180 IFPEEK(C+64)=191ORPEEK(C+64)<128THEN2190ELSEC=C+64:POKEC,17
0:POKEC-64,128:GOTO2180
2190 POKEC,128:POKEX,128:POKEX+64,128:POKEX-1,143:POKEX+1,143:PO
KEX+63,188:POKEX+65,188:A=X-1:B=X+1:C=X+63:D=X+65
2200 IFPEEK(A-65)=191ORPEEK(A-65)=179ORPEEK(B-63)=191ORPEEK(B-63
)=179ORPEEK(C+63)=191ORPEEK(C+63)<128ORPEEK(D+65)=191ORPEEK(D+65)
<128THEN2220
```

```
2210 A=A-65:B=B-63:C=C+63:D=D+65:POKEA,134:POKEA+65,128:POKEB,13
7:POKEB+63,128:POKEC,164:POKEC-63,128:POKED,152:POKED-65,128:GOTO
2200
2220 POKEA,128:POKEB,128:POKEC,128:POKED,128
2230 A=X-1:B=X+1:C=X+63:D=X+65
2240 IFPEEK(A-1)=191ORPEEK(B-64)=191ORPEEK(B-64)=179ORPEEK(C+64)
=191ORPEEK(C+64)<128ORPEEK(D+1)=191THEN2260
2250 A=A-1:B=B-64:C=C+64:D=D+1:POKEA,176:POKEA+1,128:POKEB,176:P
OKEB+64,128:POKEC,176:POKEC-64,128:POKED,131:POKED-1,128:GOTO2240
2260 POKEA,128:POKEB,128:POKEC,128:POKED,128
2270 PRINT@M,"DO YOU WANT ANOTHER GAME";:INPUTR$:IFLEFT$(R$,1)="
Y"THEN2280ELSE2730
2280 IFS>HSTHENHS=S
2290 S=0:XX=0:D4=0:C6=0:GOTO10
2300 CLS:PRINT"    YOUR ORDERS ARE TO DEFEND EARTH AGAINST THE A
LIEN FORCES    WHICH ARE NOW PREPARING TO INVADE.THEY MUST NOT SU
CCEED TO LAND THEIR GROUND FORCES AND RUN HAVOC THROUGHOUT THE WO
RLD!"
2310 PRINT"    THEY WILL INVADE IN SQUADRONS OF FIVE TROOP CARR
IERS,WHICH  DO NOT HAVE ANY SHOOTING CAPABILITIES,BUT BE WARNED,T
HEY ZIG-ZAGTOWARDS THE EARTH AND WILL DESTROY YOU IF THEY HIT YOU
."
2320 PRINT"    IF YOU ALLOW FIVE CARRIERS TO LAND YOU WILL HAVE
TO FACE THEGROUND FORCES WHICH HAVE GREATER COMBAT ABILITIES.ALSO
 EVERY    CARRIER THAT DOES LAND WILL RE-GROUP TO STRIKE AGAIN AT
 A LATER DATE."
2330 PRINT"    EVERY TIME A CARRIER LANDS OR IS DESTROYED THE OT
HER        CARRIERS IN THE SQUADRON WILL MOVE AT GREATER SPEEDS.E
VERY TIME A SQUADRON IS COMPLETELY DESTROYED YOU HAVE TO FACE A N
EW        SQUADRON."
2340 PRINT@960,"<TO CONTINUE PUSH ANY KEY>";:R$=INKEY$:IFR$<>""T
HEN2350ELSE2340
2350 CLS:PRINT"    WHEN YOU FACE THE SQUADRONS OF CARRIERS YOU W
ILL HAVE AN    ANTI-SPACECRAFT GUN TO USE AGAINST THEM,IF THIS IS
 DESTROYED YOUWILL LOSE THE GAME."
2360 PRINT:PRINT
2370 PRINT"TO USE YOUR ANTI-SPACECRAFT GUN"
2380 PRINT"################################"
2390 PRINT"TO MOVE LEFT -------   <"
2400 PRINT"TO MOVE RIGHT ------   >"
2410 PRINT"TO SHOOT  ----------   M"
2420 PRINT"TO STOP MOVING -----   ?"
2430 PRINT@960,"<TO CONTINUE PUSH ANY KEY>";:R$=INKEY$:IFR$<>""T
HEN2440ELSE2430
2440 CLS:PRINT"EVERY TIME YOU FACE A NEW SQUADRON IT WILL BE MAD
E UP OF A      DIFFERENT MODEL OF CARRIER.THE NEW MODEL WILL BE W
ORTH A        DIFFERNT SCORE AND WILL START AT A DIFFERENT HEIGHT
."
2450 PRINT
2460 PRINT"THE MODELS OF CARRIERS YOU MAY FACE"
2470 PRINT"**************************************"
2480 PRINT"CARRIER                          SCORE "
2490 PRINT"  ";:PRINTCHR$(167);:PRINTCHR$(140);:PRINTCHR$(155);:
PRINT"                              50"
2500 PRINT:PRINT"  ";:PRINTCHR$(166);:PRINTCHR$(140);:PRINTCHR$(
153);:PRINT"                              100"
2510 PRINT:PRINT"  ";:PRINTCHR$(151);:PRINTCHR$(140);:PRINTCHR$(
171);:PRINT"                              150"
2520 PRINT:PRINT"  ";:PRINTCHR$(157);:PRINTCHR$(179);:PRINTCHR$(
174);:PRINT"                              200"
2530 PRINT@960,"<TO CONTINUE PUSH ANY KEY>";:R$=INKEY$:IFR$<>""T
HEN2540ELSE2530
2540 CLS:PRINT"    THE GROUND FORCE YOU WILL FACE CONTAINS TEN A
LIENS.THESE    ALIENS WILL MOVE BACKWARDS AND FOWARDS BEHIND A BA
RRICADE       SHOOTING AT YOU.IF YOU ARE HIT BY ONE OF THEIR BULL
ETS OR YOU   RUN INTO ONE OF THEM,YOU WILL BE DESTROYED."
2550 PRINT"    ONCE YOU DESTROY THE GROUND FORCE YOU WILL BE INV
ADED BY    MORE SQUADRONS AND THE NEXT GROUND FORCE WILL TRACK YO
U SOONER."
2560 PRINT"    THE ALIENS WILL EVENTUALLY MOVE OUT FROM BEHIND T
HEIR       BARRICADES AND TRACK YOU DOWN.YOU WILL KNOW THAT THEY
ARE AFTER YOU,BECAUSE THEY WILL CHANGE SHAPE.EVERY ALIEN YOU KILL
 WILL BE WORTH FIFTY POINTS."
2570 PRINT"    YOU HAVE THE ABILITY TO SHOOT,MOVE IN ANY DIRECTI
ON AND     BEHIND THE ENEMIES BARRICADES.BEWARNED!!!---WHENEVER Y
OU DESTROYAN ENEMY ALIEN IT WILL IMMEDIANTLY SHOOT A LAST BULLET
STRAIGHT AT YOU.ALSO THEIR BULLETS CANNOT BE SHOT DOWN."
2580 PRINT@960,"<TO CONTINUE PUSH ANY KEY>";:R$=INKEY$:IFR$<>""T
HEN2590ELSE2580
2590 CLS:PRINT"YOUR CONTROLS"
2600 PRINT"XXXXXXXXXXXXX"
2610 PRINT"TO MOVE LEFT ------  <"
2620 PRINT"TO MOVE RIGHT -----  >"
2630 PRINT"TO MOVE UP --------  @"
2640 PRINT"TO MOVE DOWN ------  +"
2650 PRINT"TO STOP MOVING ----  ?"
2660 PRINT"TO SHOOT ----------  M"
2670 PRINT:PRINT
2680 PRINT"THE ALIENS"
2690 PRINT"**********"
2700 PRINT"GUARDING BARRICADES ------  ";:PRINTCHR$(158);:PRINTC
HR$(143);:PRINTCHR$(173)
2710 PRINT:PRINT"TRACKING YOU ------------  ";:PRINTCHR$(173);:
PRINTCHR$(143);:PRINTCHR$(158)
2720 PRINT @960,"<TO START PUSH ANY KEY>";:R$=INKEY$:IFR$<>""THE
NGOTO70ELSEGOTO2720
2730 CLS:FORI=15360TO15423:POKEI,153:POKEI+960,153:NEXTI:FORI=15
360TO15360+960STEP64:POKEI,153:POKEI+63,153:POKEI+1,153:POKEI+62,
153:NEXTI
2740 PRINT@472,"ALIEN INVASION";:FORI=1TO14:PRINT@535+I,"#";:NEX
T
2750 POKE15890,167:POKE15891,140:POKE15892,155:POKE16180,157:POK
E16181,179:POKE16182,174:POKE15540,173:POKE15541,143:POKE15542,15
8:POKE15690,150:POKE15691,140:POKE15692,169:POKE15585,158:POKE155
86,143:POKE15587,173
2760 PRINT@900,"TO START HIT ----- Y";:R$=INKEY$:IFR$="Y"THEN277
0ELSE2760
2770 IFS>HSTHENHS=S
2780 S=0:XX=0:D4=0:C6=0:GOTO10
```

*****WHEEL LOADER PRODUCTION L1/4K*****

```
5  CLS
8 P."==================== COPYRIGHT ===================="
10 P."GIL LYON & ASSOCIATES 306/29 YEO ST NEUTRAL BAY N.S.W."
11  P."WHEEL LOADER PRODUCTION IN TONS (2000 LBS) PER HOUR"
12  I."DO YOU WANT INSTRUCTIONS(1=YES,2=NO";U
13  CLS
23  IFU=16.25
24  IFU=26.40
25  P."THIS PROGRAMME IS DESIGNED TO CALCULATE WHEEL LOADER"
26  P."PRODUCTION IN TONS PER HOUR.IT CAN BE USED FOR ALL"
27  P."SIZES OF LOADERS IN ALL TYPES OF CONDITIONS----------"
28  P."INPUT DATA IS REQUESTED TO COVER ALL VARIABLES AND THE"
29  P."RESULT IS CALCULATED IN TONES PER HOUR.---THE PRODUCTION"
30  P."WHICH IS BASED ON AVERAGE TRUCK LOADING CONDITIONS IN"
31  P."AVERAGE DIGGING MATERIAL CAN BE USED BY CONTRACTORS FOR"
32  P."ESTIMATING JOB COSTS AND FOR SALESMEN RECOMMENDING"
33  P."AVERAGE MACHINE CAPABILITIES.------
34  P."THE TIME TO COMPLETE THE JOB CAN BE WORKED OUT AND LOADER"
35  P."SIZE AND NUMBER CAN BE USED TO ESTABLISH COMPLETION TIME"
36  P."FOR THE JOB WITH A GIVEN NUMBER AND SIZE OF MACHINE"
37  IN."TO CONTINUE,PRESS ENTER";A$
38  CLS
40 INPUT"WHAT IS THE IN BANK WT PER CU YD OF THE MATERIAL IN LBS";B
50  INPUT "WHAT IS THE BANK CORRECTION FACTOR";I
60  IN."WHAT IS THE HEAPED CAPACITY (CU/YDS) OF THE LOADER BUCKET";H
70  INPUT "WHAT IS THE EFFICENCY HOUR";E
80  INPUT "WHAT IS THE CYCLE TIME IN MINUTES";C
90  P=B*I*H*(E/C)/2000
100  PRINT P;"TONS PER HOUR"
```

*****RESTORE L2/16K*****

```
10 'THISISADUMMY.
11 DATA205,127,10,235,205,44,27,11,237,67,255,64,201
12 FORI=17136TO17148
13 READA:POKEI,A:NEXT
14 DELETE11-14
15 ' * HIGH-SPEED DATA SEARCH BY WALTER ELDRIDGE, 1 HAVEN AVENUE
 SEAFORD, S.A. 5169...(C)1981.*
```

```
20 POKE16553,255:DIM AR(25,5):DEFINT L:DEFSTR B,N,S
30 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
40 DATA0,1570,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
50 DATA1060,1090,1140,1180,1200,1220,1240,1260,1290,1310,1330,13
70,1400,1460,1490,1510,1550,1760,1790,1830,1860,1880,1900,0,1940,
0
60 DATA510,530,560,590,610,630,650,670,690,710,730,750,770,800,8
20,840,870,890,910,950,970,0,990,0,1020,1040
70 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
80 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
90 FOR C=0TO5:FOR R=0TO25:READ AR(R,C):NEXT R:NEXT C
100 CLS:PRINT@520,"";:INPUT"ENTER PLACE NAME,STATE";N,S
110 R=ASC(LEFT$(N,1))-65:C=ASC(LEFT$(S,1)):IFC=78THENC=C-77ELSEI
FC=87THENC=C-83ELSEIF(C=81)OR(C=83)OR(C=84)OR(C=86)THENC=C-81ELSE
100
120 LN=AR(R,C)
130 POKE16526,240:POKE16527,66:X=USR(LN)
140 READB:IF LEFT$(B,(LEN(N)))=N THENCLS:PRINT@520,B:GOTO160
150 IFNOTB="9999"THEN140 ELSEPRINT@960,"NOT IN THERE!!!!";
160 A$=INKEY$:IFA$=""THEN160 ELSE100
500 'TAS.(A)
510 DATA"AVOCA--003--Y","9999"
520 'TAS.(B)
530 DATA"BADEN--002--Y","BAGDAD--002--Y","BARNES WAY--002--Y","B
ARRINGTON--004--Y","BEACONSFIELD--003--Y","BEAUTY POINT--003--Y",
"BELL BAY--003--Y","BICHENO--003--Y","BISHOPSBOURNE--003--Y","BLE
SSINGTON--003--Y"
540 DATA"BOAT HARBOUR--004--Y","BOTHWELL--002--Y","BRANXHOLM--00
3--Y","BRIDGEWATER--002--Y","BRIDPORT--003--Y","BRIGHTON--002--Y"
,"BROADMARSH--002--Y","BRONTE PARK--002--Y","BUCKLAND--002--Y","B
URNIE--004--Y","9999"
550 'TAS.(C)
560 DATA"CAMPANIA--002--Y","CAMPBELL TOWN--003--Y","CARRICK--003
--Y","CASTLEFORBES BAY--002--Y","CAVESIDE--003--Y","CHRISTMAS HIL
LS--004--Y","CHUDLEIGH--003--Y","COLEBROOK--002--Y","COLES BAY--0
02--Y","CRADLE MOUNTAIN--003--Y"
570 DATA"CRADOC--002--Y","CRANBROOK--002--Y","CRESSY--003--Y","C
URRIE--004--Y","CYGNET--002--Y","9999"
580 'TAS.(D)
590 DATA"DELMONT--003--Y","DELORAINE--003--Y","DERBY--003--Y","D
ERWENT BRIDGE--002--Y","DEVONPORT--004--Y","DODGES FERRY--002--Y"
,"DOVER--002--Y","DUNALLEY--002--Y","9999"
600 'TAS.(E)
610 DATA"EAGLEHAWK NECK--002--Y","ELIZABETH TOWN--003--Y","ELLIO
TT--004--Y","EMITA--003--Y","EPPING FOREST--003--Y","EVANDALE--00
3--Y","EXETER--003--Y","9999"
620 'TAS.(F)
630 DATA"FINGAL--003--Y","FLINDERS ISLAND--003--Y","FORCETT--002
--Y","FOREST--004--Y","FORTH--004--Y","FRANKLIN--002--Y","9999"
640 'TAS.(G)
650 DATA"GARDEN ISLAND CREEK--002--Y","GEEVESTON--002--Y","GEORG
E TOWN--003--Y","GLADSTONE--003--Y","GLEN HUON--002--Y","GLENDEVI
E--002--Y","GLENGARRY--003--Y","GRETNA--002--Y","GROVE--002--Y","
GUNNS PLAINS--004--Y","9999"
```

```
660 'TAS.(H)
670 DATA"HADSPEN--003--Y","HAGLEY--003--Y","HAMILTON--002--Y","H
ERMITAGE--002--Y","HILLWOOD--003--Y","HOBART--002--Y","HUONVILLE-
-002--Y","9999"
680 'TAS.(I)
690 DATA"ILFRAVILLE--003--Y","IRISHTOWN--004--Y","9999"
700 'TAS.(J)
710 DATA"JERICHO--002--Y","JUDBURY--002--Y","9999"
720 'TAS.(K)
730 DATA"KAROOLA--003--Y","KELSO--003--Y","KEMPTON--002--Y","KET
TERING--002-Y","KIMBERLEY--004--Y","KING ISLAND--004--Y","KOONYA-
-002--Y","9999"
740 'TAS.(L)
750 DATA"LACKRANA--003--Y","LADY BARON--003--Y","LATROBE--004--Y
","LAUNCESTON--003--Y","LEBRINA--003--Y","LIENA--003--Y","LILYDAL
E--003--Y","LITTLE SWANPORT--002--Y","LONGFORD--003--Y","LUINA--0
04--Y","LYMWOOD--004--Y","9999"
760 'TAS.(M)
770 DATA"MAGRA--002--Y","MARGATE--002--Y","MARRAWAH--004--Y","MA
THINNA--003--Y","MAWBANNA--004--Y","MAYDENA--002--Y","MEANDER--00
3--Y","MELTON MOWBRAY--002--Y","MEMANA--003--Y","MIDDLETON--002--
Y","MIDWAY POINT--002--Y"
780 DATA"MILABENA--004--Y","MOLE CREEK--003--Y","MOLTEMA--003--Y
","MONTAGU--004--Y","MOOGARA--002--Y","MOORLEAH--004--Y","MORIART
Y--004--Y","9999"
790 'TAS.(N)
800 DATA"NABAGEENA--004--Y","NABOWLA--003--Y","NATIONAL PARK--00
2--Y","NATONE--004--Y","NEW NORFOLK--002--Y","NILE--003--Y","NUBE
ENA--002--Y","9999"
810 'TAS.(O)
820 DATA"OATLANDS--002--Y","ORFORD--002--Y","OUSE--002--Y","9999
"
830 'TAS.(P)
840 DATA"PAWTELLA--002--Y","PENGUIN--004--Y","PERTH--003--Y","PI
PERS RIVER--003--Y","POATINA--003--Y","PONTVILLE--002--Y","PORT A
RTHUR--002--Y","PORT HUON--002--Y","PORT SORELL--004--Y","PREOLEN
NA--004--Y","PRESTON--004--Y"
850 DATA"PYENGANA--003--Y","9999"
860 'TAS.(Q)
870 DATA"QUEENSTOWN--004--Y","9999"
880 'TAS.(R)
890 DATA"RAILTON--004--Y","RANGA--003--Y","RIANA--004--Y","RICHM
OND--002--Y","RIDGLEY--004--Y","RINGAROOMA--003--Y","ROCKY CAPE--
004--Y","ROSEBERY--004--Y","ROSS--003--Y","ROSSARDEN--003--Y","RO
WELLA--003--Y","RUNNYMEDE--002--Y","9999"
900 'TAS.(S)
910 DATA"SASSAFRAS--004--Y","SAVAGE RIVER--004--Y","SCAMANDER--0
03--Y","SCOTTSDALE--003--Y","SELBOURNE--003--Y","SHEFFIELD--004--
Y","SIDMOUTH--003--Y","SMITHTON--004--Y","SMUG--002--Y","SOMERSET
--004--Y","SORELL--002--Y"
920 DATA"SOUTH BRUNY--002--Y","SOUTHPORT--002--Y","SPRENT--004--
Y","SPRINGFIELD--003--Y","ST. HELENS--003--Y","ST. MARYS--003--Y"
,"STANLEY--004--Y","STOWPORT--004--Y","STRAHAN--004--Y","STRATHGO
RDAN--002--Y"
930 DATA"SULPHUR CREEK--004--Y","SWANSEA--002--Y","9999"
940 'TAS.(T)
950 DATA"TAKONE--004--Y","TARGA--003--Y","TARRALEAH--002--Y","TE
A TREE--002--Y","TOGARI--004--Y","TRIABUNNA--002--Y","TROWUTTA--0
04--Y","TULLAH--004--Y","TUNBRIDGE--002--Y","TUNNACK--002--Y","99
99"
960 'TAS.(U)
970 DATA"ULVERSTONE--004--Y","9999"
980 'TAS.(W)
990 DATA"WARATAH--004--Y","WATERHOUSE--003--Y","WEST PINE--004--
Y","WESTBURY--003--Y","WESTERWAY--002--Y","WEYMOUTH--003--Y","WHI
TEMARK--003--Y","WILMOT--004--Y","WINKLEIGH--003--Y"
1000 DATA"WINNALEAH--003--Y","WOODBRIDGE--002--Y","WOODBURY--002
--Y","WOODSDALE--002--Y","WYNYARD--004--Y","9999"
1010 'TAS.(Y)
1020 DATA"YAMBACOONA--004--Y","YOLLA--004--Y","9999"
1030 'TAS.(Z)
1040 DATA"ZEEHAN--004--Y","9999"
1050 'SA/NT.(A)
1060 DATA"AGERY--088--M","ALDINGA--085--A","ALFORD--088--M","ALL
ENDALE EAST--087--W","ANDREWS--088--M","ANGASTON--085--F","APPILA
--086--Q","ARDEN VALE(M)--0864871--Q","ARDROSSAN--088--M","AUBURN
(M)--088493--M"
1070 DATA"AVENUE RANGE--087--Q","AVON--088--M","9999"
1080 'SA/NT.(B)
1090 DATA"BACK VALLEY--085--F","BAGOT WELL--085--F","BALAKLAVA--
088-M","BALGOWAN--088--M","BANGHAM--087--Q","BANGOR--086--Q","BAR
MERA--085--Q","BARMERA WEST--085--Q","BAROOTA--086--Q","BATCHELOR
--089--Y","BAYLEY PLAINS--086--Q"
1100 DATA"BEACHPORT--087--W","BEAUFORT--088--M","BELALIE--086--Q
","BERRI--085--Q","BETHEL--085--F","BIRDWOOD--085--A","BISCUIT FL
AT--087--Q","BLACKFORD--087--Q","BLANCHETOWN(M)--085406--M","BLYT
H--088--M","BOOLEROO CENTRE--086--Q"
1110 DATA"BORDERTOWN--087--Q","BORDERTOWN SOUTH--087--Q","BRAY--
087--W","BRECON--087--Q","BRENTWOOD--088--M","BRIMBAGO--087--Q","
BRINKLEY--085--F","BRINKWORTH--088--M","BROOKER--086--Q","BUCKLEB
OO--086--Q","BULL CREEK--085--A"
1120 DATA"BUNDALEER NORTH--086--Q","BURDETT--085--F","BUTE--088-
-M","BUTLER BRIDGE--086--Q","9999"
1130 'SA/NT.(C)
1140 DATA"CADELL--085--M","CADGEE--087--Q","CALLINGTON--085--A",
"CALOOTE--085--F","CALTOWIE--086--Q","CANGARA--087--Q","CANNAWIGA
RA--087--Q","CANOWIE BELT--086--Q","CAPE JAFFA--087--Q","CARPENTE
R ROCKS--087--W","CEDUNA--086--X"
1150 DATA"CHINTA--086--X","CLARE--088--M","CLEVE(M)--0862811--Q"
,"COCKALEECHIE--086--Q","COFFIN BAY--086--Q","COLEBATCH--087--Q",
"COMPTON--087--W","CONMURRA--087--Q","CONMURRA SOUTH--087--Q","CO
OLTONG--085--Q","COOMBE--087--Q"
1160 DATA"COOMUNGA--086--Q","COONAWARRA--087--W","CRYSTAL BROOK-
-086--Q","CUMMINS--086--Q","CUNLIFFE--088--M","CURRAMULKA--088--M
","CURRENCY CREEK--085--F","9999"
1170 'SA/NT.(D)
1180 DATA"DALY WATERS--089-Y","DARWIN--089--Y","DELAMERE--085--F
","DUTTON--085--F","9999"
1190 'SA/NT.(E)
```

```
1200 DATA"EDEN VALLEY--085--F","EDILILLIE--086--Q","EDITHBURGH--
088--M","EIGHT MILE CREEK--087--W","EVERARD CENTRAL--088--M","999
9"
1210 'SA/NT.(F)
1220 DATA"FARRELL FLAT(M)--088433--M","FINNISS--085--A","FOUNTAI
N HEAD--089--Y","FRANCES--087--Q","FREELING(M)--085253--A","FURNE
R--087--W","9999"
1230 'SA/NT.(G)
1240 DATA"GAWLER--085--A","GAWLER RIVER--085--A","GEORGETOWN--08
6--Q","GLADSTONE--086--Q","GLENCOE--087--W","GLENROY--087--W","GL
OSSOP--085--Q","GOOLWA--085--F","GREEN PATCH--085--Q","GREENOCK--
085--F","GREENWAYS--087--W","GULNARE--086-Q","9999"
1250 'SA/NT.(H)
1260 DATA"HALBURY--088-M","HAMILTON--085--F","HAMLEY BRIDGE--085
--F","HARROGATE--085--A","HARTLEY--085--A","HATHERLEIGH--087--W",
"HILLOWN(M)--088457--M","HINDMARSH TIERS--085--F","HINDMARSH VALL
EY--085--F"
1270 DATA"HOPE FOREST--085--A","HORNSDALE--086--Q","HOYLETON--08
8--M","HYNAM--087--Q","9999"
1280 'SA/NT.(I)
1290 DATA"INMAN VALLEY--085--F","IRON BARON--086--Q","IRON KNOB(
M)--0864611--Q","9999"
1300 'SA/NT.(J)
1310 DATA"JABIRU--089--Y","JAMESTOWN--086--Q","9999"
1320 'SA/NT.(K)
1330 DATA"KADINA--088--M","KALANGADOO--087--W","KAPUNDA--085--F"
,"KATHERINE--089--Y","KATINKA--088--M","KEILIRA--087--Q","KEITH--
087--Q","KELLY--086--Q","KEPPOCH--087--Q","KEYNETON--085--F","KIM
BA(M)--0864731--Q","KINGSTON S.E.--087--Q"
1340 DATA"KINGSTON-ON-MURRAY--085--Q","KONGAL--087--Q","KONGARON
G--087--W","KOOLUNGA--088--M","KOOLYWURTIE--088--M","KOONGAWA--08
6--Q","KOONIBBA--086--X","KOONUNGA HILL--085--F","KOPPAMURRA--087
--Q","KOPPIO--086--Q"
1350 DATA"KULPARA--088--M","KYBUNGA--088--M","KYBYBOLITE--087--Q
","9999"
1360 'SA/NT.(L)
1370 DATA"LAFFER--087--Q","LANGHORNE CREEK(M)--085377--F","LARRI
MAH--089--Y","LAURA--086--Q","LIPSON--086--Q","LOCHABER--087--Q",
"LOCHIEL--088--M","LOCK--086--Q","LONG PLAINS--085--F","LOUTH BAY
--086--Q"
1380 DATA"LOWBANK--085--M","LOXTON--085--Q","LOXTON EAST--085--Q
","LOXTON NORTH--085--Q","LUCINDALE--087--Q","LUCINDALE SOUTH--08
7--Q","LYNDOCH--085--A","LYRUP--085--Q","9999"
1390 'SA/NT.(M)
1400 DATA"MAITLAND--088--M","MAKIN--087--Q","MALLALA(M)--085273-
-F","MAMBRAY CREEK--086--Q","MANGALO--086--Q","MANNANARIE--086--Q
","MARCOLLAT--087--Q","MATARANKA--089--Y","MELROSE--086--Q","MERR
ITON--086--Q"
1410 DATA"MIL LEL--087--W","MILANG(M)--085371--F","MILLICENT--08
7--W","MILTALIE--086--Q","MINGBOOL--087--W","MINLATON--088--M","M
INNIPA(M)--0868001--Q","MITCHELL--086--Q","MOCULTA--085--F","MONA
RTO--085--F","MONASH--085--Q"
1420 DATA"MONKOORA--087--Q","MOONTA--088--M","MOORAK--087--W","M
OOROOK--085--Q","MORGAN(M)--085401--M","MOUNT BENSON--087--Q","MO
UNT BURR--087--W","MOUNT COMPASS--085--A","MOUNT GAMBIER--087--W"
,"MOUNT JAGGED--085--F"
1430 DATA"MOUNT MCKENZIE--085--F","MOUNT PLEASANT--085--A","MOUN
T SCHANK--08--W","MUNDOORA--086--Q","MUNDULLA--087--Q","MUNDULLA
WEST--087--Q","MURBKO--085--M","MURRAY BRIDGE--085--F","MURRAY TO
WN--086--Q"
1440 DATA"MURTHO--085--Q","MYPOLONGA--085--F","MYPONGA--085--F",
"9999"
1450 'SA/NT.(N)
1460 DATA"NANGWARRY--087--W","NANTAWARRA--088--M","NARACOORTE--0
87--Q","NARACOORTE NORTH--087--Q","NARRIDY--086--Q","NELSHABY--08
6--Q","NEW RESIDENCE--085--Q","NHULUNBUY--089--Y","NILDOTTIE--085
--M","NORTH SHIELDS--086--Q","NURIOOTPA--085--F"
1470 DATA"9999"
1480 'SA/NT.(O)
1490 DATA"O.B. FLAT--087--W","OWEN(M)--085287--F","9999"
1500 'SA/NT.(P)
1510 DATA"PADTHAWAY--087--Q","PAGES FLAT--085--A","PALMER--085--
F","PARAWA--085--F","PARINGA--085--Q","PARUNA--085--Q","PASKAVILL
E--088--M","PATA--085--Q","PENOLA--087--W","PETHERICK--087--Q","P
INE CREEK--089--Y","PINE POINT--088--M"
1520 DATA"PINERY--085--F","PINKAWILLINIE--086--Q","POMPOOTA--085
--F","POOGINAGORIC--087--Q","PORT AUGUSTA--086--Q","PORT BROUGHTO
N--086--Q","PORT CLINTON--088--M","PORT ELLIOT--085--F","PORT GER
MEIN--086--Q","PORT JULIA--088--M"
1530 DATA"PORT LINCOLN--086--Q","PORT MACDONNELL--087--W","PORT
NEILL(M)--0868861--Q","PORT PIRIE--086--Q","PORT VICTORIA--088--M
","PORT VINCENT--088--M","PORT WAKEFIELD--088--M","PRICE--088--M"
,"9999"
1540 'SA/NT.(Q)
1550 DATA"QUORN(M)--0864871--Q","9999"
1560 'N.S.W.(B)
1570 DATA"BAAN BAA--067--Y","BAGO LOWER--069--Y","BALDERSLEIGH--
067--Y","BALDRY(M)--0636793--Y","BALFOURS PEAK--067--Y","BALLALAB
A--048--Y","BALLDALE--060--Y","BALLIMORE--068--Y","BALLINA--066--
Y","BALRANALD--(M)--0504841--W"
1580 DATA"BANAR--068--Y","BANDON GROVE--049--Y","BANGALOW(M)--06
68731--Y","BANNISTER--048--Y","BANOON--067--Y","BARADINE(M)--0684
312--Y","BARELLAN(M)--0696391--Y","BARGO--046--Y","BARHAM--054--X
","BARMEDMAN(M)--0698629--Y"
1590 DATA"BARMEDMAN EAST--069--Y","BAROOGA--058--X","BARRABA--06
7--Y","BARRY--063--Y","BARWICK--067--Y","BATEMANS BAY--044--Y","B
ATHURST--063--Y","BATLOW(M)--0694921--Y","BAWLEY POINT--044--Y","
BEARBUNG--068--Y","BEARGAMIL--068--Y"
1600 DATA"BECTRIC--069--Y","BEDGEREBONG--068--Y","BEGA--0649--Y"
,"BELBORA--065--Y","BELFORD--065--Y","BELFRAYDEN--069--Y","BELLAT
A(M)--0679372--Y","BELLBROOK--065--Y","BELLINGEN--066--Y","BELMON
T--049--Y","BELUBULA--063--Y"
1610 DATA"BEMBOKA--0649--Y","BENDALONG--044--Y","BENDEMEER--067-
-Y","BENDICK MURRELL--063--Y","BENEREMBAH--069--Y","BENI--068--Y"
,"BENTLEY--066--Y","BENWERRIN--049--Y","BERAMBING--045--Y","BERKL
EY VALE--043--Y","BERKSHIRE PARK--045--Y"
1620 DATA"BERMAGUI SOUTH--0649--Y","BEREMANGRA--062--Y","BERRIDA
LE--0648--Y","BERRIGAN--058--X","BERRIMA--048--Y","BERRY--044--Y"
,"BERRYGILL--067--Y","BERTHONG--069--Y","BETHUNGRA--069--Y","BETH
UNGRA SOUTH--069--Y","BEVENDALE--048--Y"
```

```
1630 DATA"BEXHILL--066--Y","BIALA--048--Y","BIBBENLUKE--0648--Y"
,"BIDDON--068--Y","BIDGEEMIA--069--Y","BIG SPRINGS--069--Y","BIGG
A(M)--0483511--Y","BILAMBIL--069--Y","BILBUL--069--Y","BILLIMARI-
-063--Y","BILPIN--045--Y","BIMBI(M)--0634731--Y"
1640 DATA"BINALONG--062--Y","BINDA(M)--0483551--Y","BINDOGUNDRA-
-068--Y","BINGARA(M)--0672422--Y","BINNAWAY(M)--0684412--Y","BINN
I CREEK--063--Y","BINNIA--063--Y","BIRGANBIGIL--058--X","BIRREGO-
-069--Y","BITHRAMERE--067--Y"
1650 DATA"BLACK MOUNTAIN--067--Y","BLACK SPRINGS--063--Y","BLACK
HEATH--047--Y","BLAIRMORE--067--Y","BLAKNEY CREEK--062--Y","BLAND
--063--Y","BLAXLANDS RIDGE--045--Y","BLAYNEY--063--Y","BLIGHTY--0
58--X","BOBUNDARA--0648--Y","BOCOBLE--063--Y"
1660 DATA"BODALLA--044-Y","BOGAN GATE--068--Y","BOGGABRI(M)--067
4342--Y","BOHENA--067--Y","BOMBALA--0648--Y","BONALBO--066-Y","BO
NVILLE--066--Y","BOOK BOOK--069--Y","BOOKHAM--062--Y","BOOLEROO--
049--Y","BOOMANOOMANA--058--Y"
1670 DATA"BOOMI(M)--0675342--Y","BOOROLONG--067--Y","BOOROWA--06
3--Y","BOOROWA NORHT--063--Y","BORAMBOLA--069--Y","BOURKE--068--Y
","BOWNA--060--Y","BOWNING--062--Y","BOWRAL--048--Y","BOWRAVILLE-
-065--Y","BRACKENDALE(M)--0677771--Y"
1680 DATA"BRAIDWOOD(M)--0484211--Y","BRANXTON--049--Y","BRASSI--
058--X","BRAWBOY--065--Y","BRAWLIN--069--Y","BRAYTON--048--Y","BR
EADALBANE--048--Y","BREDBO(M)--06485482--Y","BREELONG--068--Y","B
REEZA(M)--0674456--Y","BREWARRINA(M)--0687832--Y"
1690 DATA"BREWONGLE--063--Y","BRIBBAREE(M)--0638321--Y","BRIERFI
ELD--066--Y","BRIGALOW CREEK--067--Y","BRINDABELLA--062--Y","BRIN
GELLY--047--Y","BROADWATER--066-Y","BROCKLEHURST--068--Y","BROCKL
ESBY--060--Y","BROGO--0649--Y"
1700 DATA"BROKE(M)--0657921--Y","BROKEN HILL--080--W","BROOKDALE
--069--Y","BROOMS HEAD--066--Y","BRUIE PLAINS--068--Y","BRUNDAH--
063--Y","BRUNGLE--069--Y","BRUNSWICK HEADS--066--Y","BRUSHGROVE--
066--Y","BUDDAH--068--Y","BUFF POINT--043--Y"
1710 DATA"BUKALONG--0648--Y","BULAHDELAH--049--Y","BULGANDRAMINE
--068--Y","BULGARY--069--Y","BULLA CREEK--063--Y","BUNDABURRAH--0
68--Y","BUNDANOON(M)--0488351--Y","BUNDARRAH(M)0672382--Y","BUNDU
RE--069--Y","BUNGARBY--0648--Y"
1720 DATA"BUNGENDORE--062--Y","BUNGONIA--048--Y","BUNGOWANNAH--0
60--Y","BUNGWAHL--049--Y","BUNNALOO--054--X","BUNNAN--065--Y","BU
NNOR--067--Y","BUNYAH--065--Y","BURDETT--063--Y","BURINEGELL--068
--Y","BURNT YARDS--063--Y","BURRA--069--Y"
1730 DATA"BURRA CREEK--062--Y","BURRABOI--058--X","BURRAGA(M)--0
633701--Y","BURRAWANG--048--Y","BURRELL CREEK--065--Y","BURREN JU
NCTION(M)--0679612--Y","BURRINGBAR--066--Y","BURRINJUCK--062--Y",
"BURROWAY--068--Y","BURRUMBUTTOCK--060--Y"
1740 DATA"BYLONG(M)--0637983--Y","BYRON BAY--066--Y","9999"
1750 'SA/NT.(R)
1760 DATA"RAMSAY--088--M","REDBANKS(M)--085273--F","REDHILL(M)--
0863691--Q","REEDY CREEK--087--Q","REEDY CREEK SOUTH--087--Q","RE
EVES PLAINS--085--F","RENDELSHAM--087--W","RENMARK--085--Q","RHYN
IE--088473--M","RIVERTON--088473--M"
1770 DATA"ROBE--087--Q","ROOPENA--086--Q","ROSEDALE--085--A","RO
SEWORTHY--085--A","ROWLAND FLAT--085--F","RUDALL(M)--0862011--Q",
"9999"
1780 'SA/NT.(S)
1790 DATA"SANDERGROVE--085--A","SANDERSTON--085--F","SANDILANDS-
--088--M","SEBASTOPOL--087--W","SECOND VALLEY--085--F","SEDAN(M)--
085651--F","SELLICKS--085--A","SENIOR--087--Q","SEVENHILL--088--M
","SHERWOOD--087--Q","SHORT--087--W"
1800 DATA"SLEAFORD MERE--086--Q","SNOWTOWN--088--M","SOUTH HUMMO
CKS--088--M","SOUTH KILKERRAN--088--M","SOUTHEND--087--W","SPALDI
NG(M)--088453--M","STANSBURY--088--M","STEWARTS RANGE--087--Q","S
TIRLING NORTH--086--Q","STOW--088--M"
1810 DATA"STRATHALBYN--085--A","STRUAN--087--Q","SUNLANDS--085--
A","9999"
1820 'SA/NT.(T)
1830 DATA"TALDRA--085--Q","TANTANOOLA--087--W","TANUNDA--085--F"
,"TAPLAN--085--Q","TARATAP--087--Q","TARLEE--085--F","TARPEENA--0
87--W","TAYLORVILLE--085--M","TENNANT CREEK(M)--089621--Y","THORN
LEA--087--W"
1840 DATA"TILLEYSWAMP--087--Q","TINTINARA--087--Q","TOOLIGIE HIL
L--086--Q","TOOPERANG--085--A","TRURO--085--F","TUCKEY--086--Q","
TUMBY BAY--086--Q","TWO WELLS--085--A","9999"
1850 'SA/NT.(U)
1860 DATA"UNGARRA--086--Q","URANIA--088--M","9999"
1870 'SA/NT.(V)
1880 DATA"VERRAN--086--Q","VICTOR HARBOR--085--F","9999"
1890 'SA/NT.(W)
1900 DATA"WADDIKEE--086--Q","WAIKERIE--085--M","WAITPINGA--085--
F","WALKER FLAT--085--F","WALLAROO--088--M","WANDEARAH EAST--086-
-Q","WANILLA--086--Q","WARNERTOWN--086--Q","WAROOKA(M)--088554--M
","WARRAMBOO(M)--0868151--Q"
1910 DATA"WASLEYS--085--A","WATERVALE--088--M","WEETULTA--088--M
","WESTERN FLAT--087--Q","WHITWARTA--088--M","WHYALLA--086--Q","W
ILLALOOKA--087--Q","WILLAMULKA--088--M","WILLIAMSTOWN--085--A","W
ILLOW CREEK--085--F","WILLOWIE--086--Q"
1920 DATA"WILLUNGA--085--A","WILMINGTON--086--Q","WINDSOR--085--
F","WINKIE--085--Q","WIRRABARA--086--Q","WIRREGA--087--Q","WOLSEL
EY--087--Q","WOOLUMBOOL--087--Q","WRATTONBULLY--087--Q","WUDINNA-
-086--Q","WUNKAR--085--Q","9999"
1930 'SA/NT.(Y)
1940 DATA"YABMANA--086--Q","YACKA(M)--088468--M","YALLUNDA FLAT-
-086--Q","YANKALILLA--085--F","YEELANA--086--Q","YORKETOWN--088--
M","YUNDI--085--A","9999"
```

*****SOLITAIRE PROGRAM PATCH L2/16K*****

```
50 DIM V(33),C(33),P(33),N(33),Q1(33,3):Q2=1
295 IFA$="Q"THEN1000
445 Q1(Q2,1)=F:Q1(Q2,2)=T:Q1(Q2,3)=T2:Q2=Q2+1
570 PRINT:PRINT"       WISH TO MOVE. TO FINISH THE GAME,
    ENTER  'F',      TO LIST YOUR MOVES ENTER 'Q'"
575 PRINT"  TO MAKE A CORRECTION, HIT  SPACE THEN ENTER/NEWLINE"
600 IFA$="Q"THEN1000ELSE30
1000 Q$="###":CLS:FORI=1TO33:PRINTUSING"##";I;:PRINT"-";:FORJ=1T
03:PRINTUSINGQ$;Q1(I,J);:NEXT
1005 PRINT"    ";
1010 IFI/4=INT(I/4)THENPRINT
1020 NEXT
1025 PRINT@896,"ENTER 'F' TO START AGAIN, OR 'Q' TO RE-DISPLAY"
1030 POKE16537,0
1040 PRINT@768,"DISPLAY RECORDS - <MOVE NO.>  <MOVE>  <TO>  <TAK
E AWAY>";
1050 GOTO590
```

```
***** LEMNISCATES - L2/4K *****

1 I=0:J=0:INPUTK,L:P=1.57:S=.016:K=K*S:L=L*S:A=47:B=23:CLS:FORM=
0TOPSTEPS:LETI=I+K:LETJ=J+L:LETZ=SQR(COS(M)):LETX=Z*COS(I)*A:LETY
=Z*SIN(J)*B:SET(A+X,B+Y):SET(A-X,B+Y):SET(A+X,B-Y):SET(A-X,B-Y):N
EXT:GOTO1:' LEMNISCATES - C/- K. SHILLITO MAR 80
```

```
10 '********************* PUNCTUATION *********************
 (C) KEN SHILLITO                WRITTEN DECEMBER 1980

20 '********************* VARIABLES  ****************************
    A  BACKGROUND MASK   AF MASK CHARACTER        A1 WORK STRING
   B()CHANNEL MASKS      C  MIN WIDTH CHANNEL     D  MAX WIDTH CHAN
   E  DIM OF B           F  PATH WIDTH SELECTOR   G  PATH POS SLCR
30 'H  LEFT PATH LIMIT   I  FOR..NEXT CONTROL     J  FOR..NEXT CTR
L   K  FOR..NEXT CTRL    L  RIGHT PATH LIMIT      M  CONSTANT 1
   N  TORTUOSITY SLC     O  N COMPLEMENT          P  # DENSITY %
   Q  # DENSITY SLCTR    R  SCREEN ORIGION        S  ASC(#)
40 'T  ASTERISK POSN     U  ! POSN                V  SCORE KEEPER
   W  DIFFICULTY COUNT   X  MODULO FOR W          Y  ASC(*)
   Z  ! ADDRESS LIM      CC VERT TAB FACTOR       DD CONSTANT 0
   EE ASC( )             FF ASC(AF)               GG
50 'HH ! ?@ LIMIT        II KEYBOARD ADDRESS      JJ PEEK(II) FOR
,   KK PEEK(II) FOR .    LL PEEK(II) FOR /
   (A-B STRINGS - ALL OTHERS INTEGERS)
   ************************************************************
60 POKE 16612,0:CLS:PRINT"*** PUNCTUATION *#$%&'()*=+@+<>?/,.":P
RINT"YOU ARE AN ASTERISK WHICH CAN MOVE FROM SIDE TO SIDE
70 PRINT"USE THE  <  AND  >  KEYS TO MOVE SIDEWAYS
80 PRINT"OCCASIONALLY, FEARSOME # SIGNS COME TOWARDS YOU
90 PRINT"IF THEY HIT YOU, OR YOU HIT THE COLUMN WALL, YOU DIE HO
RRIBLY
100 PRINT"YOU CAN ZAP #'S WITH !'S, WHICH KILL #'S ON CONTACT
110 PRINT"TO FIRE AN ! PRESS / - BUT ONLY ONE ! CAN EXIST AT A T
IME
120 PRINT"THE PAGE GETS MORE TORTUOUS AND THE #'S PROGRESSIVELY
DENSER
130 PRINT"EVENTUALLY, YOU WILL DIE
140 PRINT"YOU WILL GET A POSTHUMOUS SCORE FOR THE #'S KILLED
150 PRINT"IF YOU SCORE WELL ENOUGH, YOU WILL RE-INCARNATE AS A <
 OR EVEN A >   - AND IF YOU'RE A REAL WHIZ, AS A CIRCUMFLEX !!!!!
160 PRINT"SIC TRANSIT GLORIA MUNDI
170 PRINT:PRINT"PRESS ANY KEY TO CONTINUE":A1$=INKEY$
180 '********** SET UP CONSTANTS **********
190 CLEAR 700:DEFSTR A-B:DEFINT C-Z
200 AF=CHR$(RND(26)+64)
210 A=STRING$(64,AF):C=20:D=30:E=2*(D-C):F=E/2:G=992-(D-C)/2:H=9
61:L=1023-D:M=1:N=3:O=(N+1)/2:P=100:Q=10:R=15360:S=35:T=15360+32:
U=0:W=0:X=40:Y=42:V=PEEK(16612):Z=16219:CC=64:DD=0:EE=32
220 FF=ASC(AF):GG=33:HH=896:II=14388:JJ=16:KK=64:LL=128
230 V=PEEK(16612)
240 IF V>10 Y=60
250 IF V>20 Y=62
260 IF V>50 Y=94
270 V=0
280 DIM B(E)
290 FOR I=0 TO D-C
300 B(I+1)=STRING$(C+I," "):NEXT
310 FOR I=D-C+2 TO E
320 B(I)=STRING$(E-I+C+1," "):NEXT
330 IF INKEY$="" THEN 330
340 I=(C+D)/2:A1=STRING$(32-I/2,AF)+STRING$(I," ")+STRING$(32-I/
2,AF)
350 CLS:FOR I=1 TO 16:PRINT A1:NEXT
360 POKE T,Y:B=INKEY$:RANDOM
370 PRINT@G,B(F);
380 '********** MAIN SEQUENCE ************
390 PRINT A;:POKE T,Y
400 PRINT@G,B(F);
410 LET F=F+RND(N)-O
420 LET G=G+RND(N)-O
430 IF F<M OR F>E LET F=M
440 IF G<H LET G=H
450 IF G>L LET G=L
460 IF RND(P)<Q POKE R+G+RND(C),S
470 IF U>Z POKE U-CC,EE:LET U=DD
480 IF PEEK(II)=LL POKE U-CC,EE:LET U=T+LL
490 IF U=DD THEN 550
500 IF PEEK(U-CC)=S LET V=V+M
510 IF PEEK(U)=S LET V=V+M
520 IF PEEK(U+CC)=S LET V=V+M
530 IF PEEK(U+LL)=S LET V=V+M
540 POKE U-CC,EE:POKE U,EE:LET U=U+CC:POKE U,EE:LET U=U+CC:IF PE
EK(U)=FF LET U=DD ELSE POKE U,GG
550 IF PEEK(II)=JJ POKE T,EE:LET T=T-M:POKE T,Y
560 IF PEEK(II)=KK POKE T,EE:LET T=T+M:POKE T,Y
570 IF PEEK(T+CC)<>EE THEN 630
580 '********** DIFFICULTY ADJUSTMENT *************
590 LET W=W+M:IF W<X THEN 390
600 N=N+2:Q=Q+5:W=DD:O=O+M
610 GOTO 390
620 '********** POST MORTEM **********
630 CLS:PRINT@17,"YOU JUST CAME TO A FULL STOP
640 PRINT@150,"YOUR SCORE WAS";V
650 FOR I=1 TO 50:PRINT@RND(192)+384,"!";:PRINT@RND(192)+384,"#"
;:PRINT@RND(192)+384,".";:PRINT@RND(192)+384,AF;:NEXT
660 PRINT@784, "WOULD YOU LIKE TO PLAY AGAIN?"
670 IF PEEK(14344)=2 THEN 690
680 IF  PEEK(14338)=64 THEN 720 ELSE 670
690 IF V>255 V=255
700 POKE 16612,V
710 GOTO 190
720 CLS:J=16445:FORI=0TO71:POKE16445,I:PRINT"GOODBYE ";:NEXT
```

***** NEXT MONTH'S ISSUE *****

Next month's issue marks the first issue in which we will also cater for the TRS-80 Colour Computer and the Hitachi Peach. An (80) after a program title indicates that the program will be for TRS-80 Model 1/3 or System 80/Video Genie computers. (Colour) indicates that the program will be for the TRS-80 Colour Computer and the Hitachi Peach.

** KEYWORD LII/16K m/l (80) **

A multi function machine language program that gives several functions in the one program, to name a few - Lazy save makes two CSAVE's with 4 second leaders, CLONE makes two cassette copies, single key entries, Get tape addresses, Hex to decimal conversion, Screen scrolling delay and lots more.

** READ A LINE LII/16K (80) **

This program was written by the author to help him locate and correct those errors that creep into your program in the early hours of the morning. Just the program for those bleary eyes.

** FLIP LII/16K (80) **

Flip is one of those habit forming games that can absorb you for hours or drive you mad in the first five minutes. Simple to play but hard to win. When you know how, it CAN be done in only seven moves.

** MULTIPLE REGRESSION ANALYSIS (COLOUR) **

This program allows multiple regression analysis between a dependent variable and two independent variables in accordance with the model Y=A+Bx1+Cx2. It calculates the coefficients for the multiple regression equation, the mean values of X1, X2 and Y, the coefficient of multiple regression and the percentage variation in Y due to X1, X2 and X1 and X2 jointly.

** SPACE COMMANDER LI/4K (80) **

You are in command of a space fleet, consisting of Battlestars and Jet Fighters, which is at war with an enemy force. The objective is to avoid capture of your Jet Fighters.

** ATOMIC TABLES (COLOUR) **

Chemistry students will find this program useful for learning their atomic tables or simply as a reference. It shows the name, symbol and element atomic number, gives the group or series and accurately gives the electron shells of an atom.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date .....

To **MICRO-80**

Please consider the enclosed program for . . .

Tick where appropriate

(i) Publication in MICRO-80 .....

(ii) Publication on disk or cassette only .....

(iii) Both .....

Name .....

Address .....

.....

.....Postcode.....

*** * * CHECK LIST * * ***

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE/DISK EDITION INDEX *****

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. The cassette also contains the source code for machine language programs which may not have been printed due to space restrictions. All programs are recorded twice. Level I programs can only be loaded into a Level I TRS-80 unless the Level I in Level 2 program from the MICRO-80 Software Library - Vol. 1 is first loaded into your Level 2 TRS-80 or System 80/Video Genie. Note: System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

The disk edition contains all those programs which can be executed from disk, including Level I programs. Level I disk programs are saved in the NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

	TYPE	I.D.	DISK FILESPEC	APPROX. START POSITION CTR-41	CTR-80	SYSTEM-80
SIDE 1						
LEMNISCATES	L2/4K L	L	LEM/BAS	18	10	10
"	"	"	"	27	15	16
ALIEN INVASION	L2/16K	A	ALIEN/BAS	36	20	21
"	"	"	"	169	95	100
RESTORE	L2/16K	R	RESTORE/BAS	284	160	168
SIDE 2						
PUNCTUATION	L2/16K	P	PUNCUATE/BAS	18	10	10
"	"	"	"	61	34	36
SOLITARE PATCH	L2/16K	S	SPATCH/BAS	100	56	59
"	"	"	"	111	62	65
WHEEL LOADER PRODUCTION	LI/4K	-	LOADER/LVI	120	68	71
"	"		"	149	84	88
RESTORE	L2/16K	R	RESTORE/BAS	176	99	104

A SPECIAL OFFER!

TO SUBSCRIBERS OF MICRO 80

FOR THE FIRST TIME IN AUSTRALIA!
GENUINE TANDY TRS–80 MICRO-COMPUTERS, HARDWARE AND SOFTWARE AT

GREAT DISCOUNT PRICES!!!

HOW TO ORDER –

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL –

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS *
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

* *subject to availability*

to CONQUEST ELECTRONICS Pty. Ltd.
212 Katoomba St. KATOOMBA 2780

Please supply –

QTY.	CAT NO	DESC.	ADV. PRICE

SUB TOTAL _____
LESS 10% _____
FIND CHEQUE FOR **TOTAL** _____

SEND **FREIGHT FREE** TO

NAME .....
ADDRESS .....
.....P/ Code .....

TANDY ELECTRONICS DEALER

CONQUEST ELECTRONICS PTY. LTD.

212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491

MICRO-80

LEVEL 2 ROM
ASSEMBLY LANGUAGE TOOLKIT
by Edwin Paay
FOR TRS-80 MODEL 1, MODEL 3
AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.
- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements etc. It also describes the various formats used for BASIC, System and Editor/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory — the only restriction is your own imagination!

The Appendices contain sample programmes which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p

— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...

UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!

Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT

MICRO-80